13 HSM Command-line Tasks

Generally, HSM is configured from the graphical interface and requires very little day-to-day maintenance. This chapter describes the non-routine configuration and maintenance tasks that are performed mostly from the command line.

- HSM Commands
- Test Staging
- Set Up Periodic Staging
- Tuning for Staging to Tape
- Coordinating Automatic Procedures
- Working with Individual Files
- Checking the Staging Configuration
- Copying and Moving Data
- Monitoring Storage Space and File Sizes
- Maintaining Non-Stageable Filesystems
- Managing Your Magnetic Disks
- Populating Filesystems
- Disabling Filesystem Staging
- Compacting Staging Media

- Compacting Baseline Media
- Clearing Incomplete Bitfiles
- Gathering Migration Store Statistics
- Checking a Network Client's Staging Configuration
- Troubleshooting HSM
- Restoring a Lost or Damaged Staging Volume
- Restoring a Lost or Damaged Staging Trail
- Restoring a Lost or Damaged Filesystem

HSM Commands

A brief description of the administrative and configuration commands available for local and network HSM systems can be found in "HSM Man Pages" on page 18-9.

Test Staging

Test that staging works by copying a large file, such as /etc/termcap, to a stageable filesystem, stage it out, and verify that staging took place. Refer to the corresponding man pages for details concerning command arguments.

```
emc# cd /home1
emc# cp /etc/termcap
emc# emstage termcap
emc# emls

 Mag KB Stg KB   Staging media       Filename
      8      0                       lost+found
      8     87   #0155-a doc         termcap
```

Set Up Periodic Staging

Make sure an entry exists in root's crontab file for scheduling nightly, periodic staging runs. The following example is inserted into root's crontab file by the HSM installation procedure.

```
00 0 * * * /bin/kill -HUP `cat /usr/epoch/etc/msl/emmasterd.pid` >/dev/null 2>&1
```

Minutes

Hour

Tuning for Staging to Tape

EDM with HSM option comes configured for staging to optical media. If you are staging to tape, you can improve performance by causing larger stage-ins, and thereby less tape repositioning.

You do this by using two parameters in the server's /usr/epoch/etc/msl/msl.cfg file and then enabling this modified configuration file.

Optical media has a relatively quick *seek time* when compared to tape. Therefore, a small amount of data can be efficiently staged in from optical. This is how HSM is configured by default.

For tape, however, reading small amounts of data for multiple users can cause thrashing against the tape drive. Larger stage-ins are required to cause less tape repositioning. The larger stage-ins are possible because magnetic media has a faster transfer rate than optical and a much greater storage density.

Stage-to-Tape Tuning Parameters

The parameters are:

- MSP_READ_AHEAD_PERCENTAGE

  The percentage of the file to be read on each read-ahead during a stage-in. It must be a whole number.

- MSP_READ_AHEAD_MAX

  A throttle on MSP_READ_AHEAD_PERCENTAGE, it is the maximum number of bytes that are read on any give read-ahead. (This parameter is used to prevent the stage-in of a large file from taking over the system.)

The default settings for optical are:

```
MSP_READ_AHEAD_PERCENTAGE=25
MSP_READ_AHEAD_MAX=1048576
```

If you are staging in from magnetic tape and have concurrent users you may want to change these to:

```
MSP_READ_AHEAD_PERCENTAGE=34
MSP_READ_AHEAD_MAX=62914560
```

For example, if a user tries to read in a 1 gigabyte file and the MSP_READ_AHEAD_PERCENTAGE is 34, then HSM tries to read in 300 megabytes.

If you set MSP_READ_AHEAD_MAX=62914560, the stage-in is limited to 60 megabytes (or about 1 minute of time).

Stage-to-Tape Tuning Procedure

To edit these parameters and enable the modified configuration file:

1. Edit the /usr/epoch/etc/msl/msl.cfg file to edit or add the two parameter values.

   Note: You should make a copy of the original /usr/epoch/etc/msl/msl.cfg file.

2. Kill an emfmd daemon (killing one child process should kill them all).
3. Do an **init 6**

   or

Restart the emfmd daemons with:

```
    /usr/epoch/lib/msl/emfmd
```

Coordinating Automatic Procedures

To ensure maximum efficiency, you should coordinate all of the automatic backup and staging procedures that are run through root's crontab file. Table 13-1 shows two sets of recommended order of tasks: one for systems with only local HSM and backup and the other for systems with local and network HSM and backup.

Table 13-1 **Recommended Order of Procedures in crontab**

System with local HSM and backup	System with local and network HSM and backup
	1. Client: periodic stage out
1. Server: **emvck**	2. Server: **emvck**
2. Server: periodic stage out	3. Server: periodic stage out
3. Server: compaction	4. Server: compaction
4. Server: baseline backup (optional)	5. Server: baseline backup (optional)
5. Server: regular backup	6. Server: regular backup
	7. Client: backup
	8. Server: database backup

The remainder of this section provides additional detail for these activities.

Backup automatically performs baseline backups before regular backups, and in sites with HSM clients, perform a database backup last. You only need to schedule client backups (step 7) after the server backup (step 6) in sites with HSM clients.

Client (Periodic Stage Out)

It is always best to stagger your network client's periodic staging runs so that their activity does not overload the network or the server. If, however, you stage multiple clients simultaneously, performance improves if the clients stage to different server disks.

Note: You should configure a larger delay factor (closer to twenty minutes) for older and slower hardware with large files or a large numbers of files.

If a client system has more than one stageable filesystem, stagger the staging of each filesystem by 12 to 20 minutes. It is especially important to set this delay factor on filesystems that contain large files of more than one MB. Running more than one staging operation simultaneously can degrade throughput.

The following entry in root's crontab file starts a new staging day at 12:15 every night:

```
15 0 * * * /bin/kill -HUP 'cat /usr/epoch/etc/mal/emmasterd.pid' > /dev/null 2>&1
```

Refer to "Periodic Staging and Filesystem Delay" on page 11-22 for further information.

Server (emvck)

The **emvck** (volume check) program reads filesystem information on magnetic disk and compares it to the database. If the results for a staging volume do not match, it generates accurate counts, updates the database, and logs a message. The following entry in root's crontab runs **emvck** at 11:45 every night:

```
15 23 * * * PATH=/usr/epoch/bin:$PATH;export PATH;emvck >/dev/null 2>&1
```

Server (Periodic Stage Out)

Schedule nightly periodic staging of all client stores. Stagger the staging so that the activity does not overload the optical library unit.

Server (Compaction)

Schedule nightly compaction of staging volumes by running **emcompact** every night from root's crontab file. After the source volumes are compacted, they are made available for reuse.

Check the message files every day. If automatic compaction frequently fails to reach the free goal, the system may have reached its limit, given the number of volumes in the library unit. Use the **dbreport compaction** report to determine whether there is enough stale space to reclaim, or whether you should add more disks to your library unit. Refer to "Compacting Staging Media" on page 13-28 for further details.

Server (Baseline Backup)

Baseline backups are run automatically before regular backups. The recommended baseline backup procedure is to have backup templates use both primary and alternate trailsets, with both trailsets specifying a single baseline trail.

You can back up the server, several clients, and the server database all within a single backup template.

Server and Client (Backup)

ebbackup should back up the server first and then your clients. If you are running network HSM, it is important to back up the client stores on the server, *before* you back up the clients. The following crontab entry starts a backup at 10:30 each night, using the backup template called *default*:

```
30 22 * * * /usr/epoch/EB/bin/ebbackup default > /dev/null 2>&1
```

Refer to Chapter 14 "Start of Backup and Related Processing" for more information.

Server (Backup Database)

Always back up the backup databases on the server after you back up the clients. This is also the case even if you have no network clients, because the server is considered a "local" client to the backup software. Database backups provide you with complete information about both the server and client backups and shorten disaster recovery time because they enable you to restore the database independently from the files that were already backed up.

By default, the backup software ensures that the backup database is backed up last.

Rotating Error Logs

Log files reside in a non-stageable filesystem in the /var/adm/epoch directory. To prevent the log files from growing excessively, the **epnewlog** script, which is run weekly from root's crontab file, will rotate or archive a log file to a stageable filesystem (/usr/epoch/adm/rotated) whenever the log file grows to more than one megabyte. The script rotates the concise log and the mntfault log using the usual Unix-style rotation scheme (*.0, *.1, *.2,...):

The script archives the detail log permanently to /usr/epoch/adm/archived using a date-based suffix.

epnewlog does the following:

1. Moves each message log file from the /var/adm/epoch directory to the /usr/epoch/adm/rotated directory. For example, the message log file /var/adm/epoch/concise is moved to /usr/epoch/adm/rotated/concise.
2. Makes each rotated log file in /usr/epoch/adm a version.0 file. For example, the message log file shown in step 1 becomes /usr/epoch/adm/rotated/concise.0. Each time **cron** runs the **epnewlog** script, the file suffix is incremented (.1,.2,.3) and finally deleted.
3. Assigns a date-based suffix to the detail log file in /usr/epoch/adm/archived.
4. Creates new message files in the /var/adm directory.
5. Restarts the **syslogd** process.

For the rotated log files, this procedure saves an entire month's worth of log data by rotating the log file names: messages.0 –> messages.1 –> messages.2 –> messages.3 –> deleted. Because the messages.[0–3] log files are in the /usr filesystem and are no longer growing, they are now candidates for HSM to staging devices. Use a similar approach if you've configured any other log files.

Working with Individual Files

The HSM Configuration interface handles data among several layers in a storage hierarchy. However, the software also provides command-line tools for manipulating individual files. The following procedures describes these tools:

- Staging out files
- Staging in a set of files
- Tagging a set of files for future stage out
- Locking a file on magnetic disk

Staging Out Files

During normal system use HSM manages staging for you. There are times, however, when you know that a group of files is no longer needed. For example, when you finish a project, you can explicitly stage those files out.

Use the **emstage** command to stage out or prestage one or more files. The following example stages out the files, drawing1 and drawing2:

```
emc% emstage drawing1 drawing2
```

There are command line arguments that let you prestage files, to recursively descend subdirectories while staging, to follow symbolic links, to stage to a particular volume, and to stage to a volume that contains a particular file. See the **emstage** man page for details.

Staging In a Set of Files

If you know that a group of staged-out files are needed soon, you can use the **embsi** command to stage them in. The **embsi** command stages the files in and attempt to make them all resident on magnetic disk.

By default, **embsi** does not attempt to stage in files if there is not enough available space for them on magnetic disk. Also, by default, **embsi** does not update access times. To force **embsi** to stage in files even if not enough magnetic disk space is available, use the **-f** option; to update access times, use the **-a** option.

When access times are not updated, **embsi** estimates available free space; when access times are updated, **embsi** estimates the total disk space in the filesystem. If not enough space is available, **embsi** displays the amount required and its estimate of the amount available. You can select a smaller set of files to stage in, explicitly stage out files, or force the stage in.

Forcing a bulk stage in, when there is not enough space, almost certainly means that HSM will have to stage out some files. If the access times are updated, the chances are that many of the files that are being staged in remain on magnetic disk and other files are staged out. If the access times are not updated, the chances are that at least some of the files being staged in do not remain on magnetic disk.

The following example recursively stages in all the files in the current directory:

```
emc% embsi -r .
```

Tagging a Set of Files for Future Stage Out

Use **emchmod -C** to tag files so that they are staged out at the next convenient time, which is usually the next periodic staging run. Remember that, **emchmod**, unlike **chmod**, clears properties if they are not specified on the command line. Therefore, you should always use **emls -l** first to determine the properties that are already set. You must be superuser or the owner of the file(s) to change properties.

To tag a set of files for future stage out:

1. Change to the directory that contains the files you want to stage out.

   ```
   emc# cd archive
   ```

2. Use **emls -l** to determine the properties that are already set.

```
emc# emls -l *
```

```
Mag KB Stg KB  I-flags    Flags     Staging media   Volume barcodes Filename
  1024      0  ----   0   l--- 60          -         ----           filexyz
    24    898  ----   0   ----  0   #002-a  Archive  ----           fileabc
     1      0  --CK  60   ----  0          -         ----            dirabc
```

3. Use **emchmod** with the **-C** option to indicate that the specified files should be staged out at the next convenient time. (The **-P** option sets the residence priority.)

```
emc# emchmod -C -P36 filexyz
```

Locking a File on Magnetic Disk

Use **emchmod -l** to lock file(s) on magnetic disk and prevent the files from being staged out. Remember that, **emchmod**, unlike **chmod**, clears properties if they are not specified on the command line. Therefore, you should always use **emls -l** first to determine the properties that are already set. You must be superuser or the owner of the file(s) to change properties**.**

To lock a file on magnetic disk:

1. Change to the directory that contains the subdirectory or files you wish to lock on magnetic disk.

```
emc# cd archive
```

2. Use **emls -l** to determine the properties that are already set.

```
emc# emls -l *
```

```
Mag KB Stg KB  1-flags   Flags     Staging media   Volume barcodes Filename
  1024      0   ---- 0    ----60           -         ----            filexyz
    24    898   ---- 0    ---- 0   #002-a  Archive   ----            fileabc
     1      0   --CK 60   ---- 0           -         ----             dirabc
```

3. Use **emchmod** with the **-l** option to lock the specified files on magnetic disk.

```
emc# emchmod -l filexyz
```

Consider locking down VxFS reserved files. Reserved files are configured on VxFS to remain on the magnetic disk, so they are good candidates for locking. See the VxFS documentation for information on reserved files.

Locking too many files can seriously impact HSM performance. Before you lock files onto magnetic disk, try small changes to the residence priority (see the **emchmod** man page). Be careful about setting the inheritable lock property on a directory. All files and directories created below that point inherit the lock property.

Checking the Staging Configuration

Use the **emcheck** command to check the staging configuration. You must be superuser to run **emcheck**. If you type the command without any arguments it checks the configuration database, warns you of potential problems and corrects inconsistencies. If you use the **-v** switch (verbose) you see more information. If you use the **-r** switch (read-only), **emcheck** does not correct any problems that it may find.

The **emcheck** command checks the database in eight phases. The first four phases, D1-D4, verify the semantics and the syntax of the configuration database. The second four phases, S1-S4, verify the system as a whole.

The **emcheck** command displays the following lines:

```
beta# /usr/epoch/bin/emcheck
```

```
*** Phase D1 - Checking configuration database files
*** Phase D2 - Checking status of Epoch Migration servers
*** Phase D3 - Checking store database against server config
*** Phase D4 - Checking status of stores
*** Phase S1 - Check Epoch Migration directories
*** Phase S2 - Check for running Epoch Migration daemons
```

Copying and Moving Data

With HSM you have the ability to store vast amounts of data throughout your network. Often you may find the need to copy or move data from one location to another – between filesystems, network clients, or servers, for example. This section provides step-by-step instructions for copying and moving large amounts of data between various locations in your network. The topics are:

- Migrating data from one staging trail to another
- Copying files from one filesystem to another
- Moving and copying files between HSM systems
- Copying files between network HSM clients
- Copying files to a non-EDM system

Migrating Data from One Staging Trail to Another

After you've had your system for a while, you may want to archive some older data to tape or optical disk. To do this, use the **restage** command on the EDM server to migrate files from one staging trail to another.

1. Decide which staging trail you want to restage the data to.
2. If necessary, create a new staging template/trail with the **emstconf** command.
3. Use the **restage** command to migrate the data from the existing staging trail to the new one.

   The following command moves staged files from the doc trail to the tape1_trail. The command moves only those files that haven't been modified or accessed in the last 30 days.

```
# restage -t tape1_trail -R /data1 -mtime +30 \ -atime +30 -staged_to_trail doc
```

See the **restage** man page for additional examples.

Copying Files from One Filesystem to Another

The fastest, most efficient way to move or copy large amounts of data from one filesystem to another is with the **ebcp** command. The **ebcp** command copies files from one place to another within an HSM-enabled system, between HSM-enabled systems, or from an HSM-enabled system to a non-HSM system.

The **ebcp** command automatically determines whether the destination filesystem is under HSM control. If so, **ebcp** copies the files that are not staged out to the new location. In addition, **ebcp** will create a filesystem entry for staged-out files and attach these entries to their staged images. If the destination location is not under HSM control, **ebcp** will copy everything, including the files that are staged out.

When copying to a filesystem that is not under HSM control, you must be sure that enough space is available to hold all of the files that are copied.

The following example copies all of the files in the /data1 filesystem to the /data2 filesystem on the same HSM-enabled system:

1. Change to the source directory:
   ```
   emc# cd /data1
   ```
2. Use **ebcp** to copy the files:
   ```
   emc# ebcp . /data2
   ```

After you copy the files, you can delete the originals.

Moving and Copying Files Between HSM Systems

There are two general approaches to copying files between HSM-enabled systems. Both approaches use the **ebcp** command. In the first approach, you simply copy all of the files and directories from one fileserver to another, including everything that is staged out, without transferring the staging media to the new fileserver.

In the second approach, you copy only the magnetic-resident data to the new fileserver and then reattach to the staging media. This second approach is significantly faster and is especially recommended when the quantity of data is too great to transfer over the network.

Copying Files to Another HSM System (No Media)

The following example copies all of the files, including the files that are staged out, in the /data1 filesystem on a system named "emc" to the /data2 filesystem on a system named "emc2." This example will work on any system configured for HSM (fileservers and clients).

```
# ebcp -o /data1 | rsh emc2 /usr/epoch/bin/ebcp \ -i /data2
```

Note that **ebcp** does not update directory modification and access times, unless you issue the command a second time, using the **-dir** switch. See the **ebcp** man page for details.

Moving Files to Another EDM (Media Included)

The following example copies all of the files that are not staged out from the /data1 filesystem on a system named emc to the /data2 filesystem on a system named emc2. This example uses the **-local** switch so that **ebcp** reattaches files to the imported staging media rather than copy the staged-out files.

1. Insert all of the source machine's staging media into the inlet of the destination machine's library unit. *Do not* press any of the buttons on the front of the library unit.
2. Use the Library Unit Manager to import all of the volumes.
3. Run **ebfs_import -a** to complete the import.
   ```
   emc2# /usr/epoch/bin/ebfs_import -a
   ```
4. Use **ebcp** to copy the files that aren't staged out to the new system and to reattach staged-out files to their staged images. Note that this example only works with the primary staging media. The secondary staging media (the baseline backups) cannot be moved. (The **-R 1S** switch, which tells

ebcp not to copy the baseline information, is actually an option to **recxcpio**. See the **recxcpio** man page for details.)

```
emc# ebcp -o -local /data1 | rsh emc2 \ /usr/epoch/bin/ebcp -i /data2 -R 1S
```

5. After the copy has completed, you can delete the files on the source machine.

Moving Files Between HSM Clients (Store Included)

It is also possible to copy a large set of files, or an entire filesystem, from one network client to another. You can use **ebcp** to copy the magnetic-resident files to the new client and simply change the ownership of the client store from one system to another.

Note: If other filesystems are staging to the store, you need to repeat this procedure for each filesystem.

The procedure is as follows:

1. Bring the source filesystem to an inactive state.
2. On the EDM, use **emschs** to change the ownership of the client store from the source to the destination client.

   The following command changes the ownership of the alpha_all store to a network client named beta:

   ```
   emc# emschs alpha_all -c beta
   ```

3. On the EDM, use **emsmvs** to change the name of the store from alpha_all to beta_all:

   ```
   emc# emsmvs alpha_all -n beta_all
   ```

4. If necessary, create a staging template on the destination system and configure the destination filesystem for staging.
5. Copy all the files from the data1 filesystem on the network client named alpha to the /data1 filesystem on the network client named beta:

```
alpha# ebcp -o -local /data1 | rsh beta \ /usr/epoch/bin/ebcp -i /data1 -R 1S
```

6. Repeat the procedure for all other client filesystems that stage to the same client store.

7. After the copy completes, delete the files on the source machine.

Restoring a Staged Out File That Has Been Deleted

When a file is staged out from an HSM client to a client store, the bitfile gets backed up by the fileserver's backup, and the file attributes that remain on the client get backed up by the client backup. If someone deletes the file, both the attributes on the client and the bitfile on the fileserver are deleted. You will not be able to access the file's data until the attributes have been restored on the client and the bitfile has been restored on the fileserver.

To restore the staged-out file:

1. On the client run **edmrestore** to start the EDM Restore window.
2. Select the client and work item.
3. Select the deleted file from the list.
4. Verify the destination and start the restore.
5. Run **emsundel** on the EDM server to restore the bitfile.

   ```
   emc# emsundel
   ```

   emsundel by default restores all deleted stores that have been recovered. The command should be run regularly from an entry in the root's crontab file. If you can wait, let the **emsundel** crontab entry restore the bitfile. The **emsundel** man page describes how restrict **emsundel** to a single store and force it to use a specific backup template.

Copying Files to a Non-EDM System

It is also possible to use **ebcp** to copy data from an HSM-enabled system to any Unix system. This example copies all files in the /data1 filesystem, including the files that have been staged out, on a system named "emc" to the /tmp/output filesystem on a system named "colt."

1. Change to the source directory:

```
emc# cd /data1
```

2. Mount the destination filesystem on your EDM:

```
emc# mount colt:/tmp/output /mnt
```

3. Use **ebcp** to copy the files:

```
emc# ebcp . /mnt
```

4. After the copy has completed, you can delete the files in the source filesystem.

Monitoring Storage Space and File Sizes

HSM provides several tools that enable you to monitor storage space and file sizes. These tools are listed in Table 13-2. See the man pages for further information.

Table 13-2 **Monitoring Commands**

If you want to:	Use this command:
Show magnetic disk space used by directories.	**emdu**
Show magnetic disk space used by directories and individual files.	**emdu -a**
Show virtual storage space used by staged-out directories and files; also shows magnetic disk space used by magnetic-resident directories and files.	**emdu -av**
Find out what staging volume a file resides on and show file sizes on magnetic disk and staging media.	**emls**
Show information about used and available disk space for each filesystem. (Syntax and display may vary from system to system.)	**df**
Show information about used and available inodes for each filesystem. (Syntax and display may vary from system to system.)	**df -g**

Table 13-2 **Monitoring Commands (Continued)**

If you want to:	Use this command:
List all the volumes in a staging trail and obtain information about current staging volumes.	**dbreport appl_usage**
Determine which staging volume to compact.	**dbreport compaction**
Display virtual filesystem statistics.	**emfsreport**

Maintaining Non-Stageable Filesystems

The root filesystem and the filesystem that contains the EDM software cannot be configured for HSM because they contain files that must not be staged out. These and any other non-stageable filesystems can therefore fill up.

The most likely reason that these filesystems would fill up is some sort of unexpected activity, either accidental or deliberate. If a filesystem becomes full, find files that can be deleted and determine why usage is increasing. Do the following to determine the cause of the problem:

- Use **ps** to determine what processes are running and kill any unexpected ones.
- Look at /tmp and /usr/tmp and delete any unnecessary temporary files.
- Use **find** to look for new files in the root filesystem.
- Examine the HSM log files in /usr/epoch/adm/archived.

Managing Your Magnetic Disks

If a filesystem is constantly staging files in and out, it may have an inadequate *working set*. The working set is the amount of stageable magnetic data that a filesystem can hold without exceeding its low watermark.

The working set is often measured as the number of days worth of files that are stored on the magnetic disk. This is called the *working set in days*. For most non-archival applications, you should have a working set period of at least one to four weeks. If your working set of files is considerably larger than your actual magnetic disk space, you will experience system performance degradation.

Use the **emfsreport** tool to display virtual filesystem statistics and to develop a coherent strategy for managing your magnetic disks:

1. Run **emfsreport** to display virtual filesystem statistics.
2. Decide how many days worth of data you need to keep on the magnetic portion of a filesystem.
3. Determine the additional magnetic disk space you'll need.
4. Reconfigure your magnetic disk usage or purchase more disks.

These steps are described in detail below.

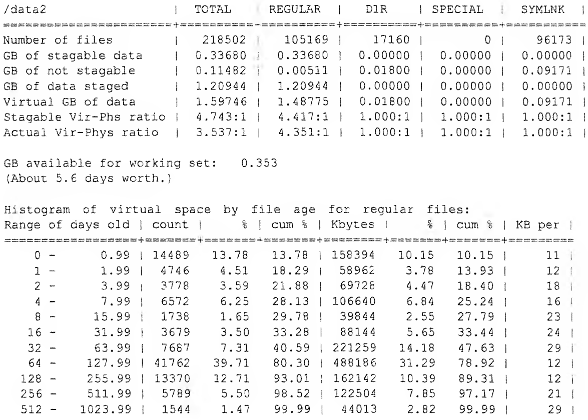
Run emfsreport

Use **emfsreport** to find out your *working set size* and the *working set in days*, so that you can fine tune your system. To run **emfsreport**, become root and specify either the name of a locally mounted filesystem, or use the **-a** switch for all filesystems. For example:

```
emc# emfsreport -hva /data2
```

A sample report is shown in Figure 13-1. It displays the amount of space used by all files in the /data2 filesystem, regardless of their location. (Note that virtual space takes into account space consumed on magnetic disk plus space consumed on the staging media.)

Figure 13-1 **emfsreport Output**



```
/data2                  |  TOTAL    REGULAR  |   DIR     | SPECIAL  |  SYMLNK  |
========================+==========-==========+==========+==========+==========
Number of files         |   218502 |   105169 |    17160 |        0 |    96173 |
GB of stagable data     |  0.33680 |  0.33680 |  0.00000 |  0.00000 |  0.00000 |
GB of not stagable      |  0.11482 |  0.00511 |  0.01800 |  0.00000 |  0.09171 |
GB of data staged       |  1.20944 |  1.20944 |  0.00000 |  0.00000 |  0.00000 |
Virtual GB of data      |  1.59746 |  1.48775 |  0.01800 |  0.00000 |  0.09171 |
Stagable Vir-Phs ratio  |  4.743:1 |  4.417:1 |  1.000:1 |  1.000:1 |  1.000:1 |
Actual Vir-Phys ratio   |  3.537:1 |  4.351:1 |  1.000:1 |  1.000:1 |  1.000:1 |

GB available for working set:   0.353
(About 5.6 days worth.)

Histogram  of  virtual  space  by  file  age  for  regular  files:
Range of days old | count |     % | cum % | Kbytes |      % | cum % | KB per |
==================+=======+=======+=======+========+=======+=======+========
    0 -      0.99 | 14489   13.78   13.78 | 158394   10.15   10.15 |     11 |
    1 -      1.99 |  4746    4.51   18.29 |  58962    3.78   13.93 |     12 |
    2 -      3.99 |  3778    3.59   21.88 |  69728    4.47   18.40 |     18 |
    4 -      7.99 |  6572    6.25   28.13 | 106640    6.84   25.24 |     16 |
    8 -     15.99 |  1738    1.65   29.78 |  39844    2.55   27.79 |     23 |
   16 -     31.99 |  3679    3.50   33.28 |  88144    5.65   33.44 |     24 |
   32 -     63.99 |  7687    7.31   40.59 | 221259   14.18   47.63 |     29 |
   64 -    127.99 | 41762   39.71   80.30 | 488186   31.29   78.92 |     12 |
  128 -    255.99 | 13370   12.71   93.01 | 162142   10.39   89.31 |     12 |
  256 -    511.99 |  5789    5.50   98.52 | 122504    7.85   97.17 |     21 |
  512 -   1023.99 |  1544    1.47   99.99 |  44013    2.82   99.99 |     29 |
```

Choose the Desired Working Set in Days

Using the values displayed in Figure 13-1, you can calculate how much more magnetic space you would need for a desired working set size in days. Figure 13-1 shows that /data2 has 353 MB available in its working set, with a working set in days of 5.6 days. Once you learn what your working set size in days is, you may decide that it is too small. (Remember, EMC recommends you have a working set period of at least one to four weeks.) Select an adequate number of days based on your usage patterns. See "emfsreport and the Working Set" on page 11-31 and "Disk Utilization Zones" on page 11-13 for some additional considerations.

Determine Additional Magnetic Disk Space Required

The sample report of /data2 shows that the filesystem has a working set of 5.6 days. Suppose that is unacceptable and you'd rather have a working set of 14 days. How much additional magnetic disk space would you need?

To estimate, simply pick the high day range that falls closest to your desired working set size. In this case the day would be 15.99.

1. Multiply the value on the "Kbytes cum%" line (27.79) by the total number of Kbytes (1560023) to get the working set size in KB. For example:

   Working set size in KB = .2779 * 1,560,023

   The result is 433530.39 KB.

2. Take the result of the previous calculation (433530.39) and divide by 1024 to get the working set size in MB.

   Working set size in MB = 433530.39/1024

   The result is 423.37 MB.

3. Take the result of the previous calculation (423.37) and divide by 1024 to get the working set size in GB.

   Working set size in GB = 423.37/1024

   The result is 0.41 GB.

4. Subtract the GB available for the working set (0.353) from 0.410 and multiply by 1024 to get the number of additional magnetic space needed in MB.

   Mag Disk Space needed = (.410 - .353) * 1024

   The result is 58.37 MB.

Thus you would need approximately 58.37 Mbytes more magnetic disk space on this filesystem in order to have a working set of 15.99 days. (You can extrapolate to find the exact requirements for a working set size of 14 days.)

Reconfigure Magnetic Disk or Purchase More Disks

Depending on your site's configuration, your budget, and the amount of time you have, there are a number of actions you can take to better utilize your magnetic disk resources:

- Move files to another filesystem that is under utilized. This filesystem may be on the same or on another magnetic disk.
- Repartition your disk. Take a complete backup of your filesystems, repartition your disk and restore your files. When repartitioning your disk, make the partitions correspond to their working set needs; make some partitions smaller and some larger.
- Add magnetic disks if you are using all of your current disk space and have no where else to relocate the files.

Populating Filesystems

The most efficient way to move files from a non-HSM system to an HSM system is to make the HSM system an NFS client of the other system and to use **tar** to read the data. For example, to copy /user1 on another system to /data1/user1 on the HSM system, use the following procedure:

1. Configure the other system as an NFS server for its /user1 filesystem.
2. Login to the HSM system as root, make a temporary directory, and temporarily mount the server's /user1 as a remote filesystem:

```
emc# mkdir /other_user1
emc# mount -o,ro,hard,intr serv:/user1 /other_user1
```

3. Change to the remote filesystem and use **tar** in a pipeline to copy files to the new filesystem on the HSM system. Then, unmount the remote filesystem and delete the temporary directory:

```
emc# mount -F vxfs -o remount,nolog /data1
emc# cd /other_user1
emc# tar cBf - .|(cd /data1/user1;tar xBpf -)
emc# cd /
emc# umount /other_user1
emc# rmdir /other_user1
```

An HSM-enabled system can also be populated by configuring it as an NFS server, configuring the other system as the client, and using **tar** on the client system to "push" the files to the HSM system. Because NFS read performance is generally better than NFS write performance, the best approach is to make the HSM system the client and "pull" the files from the server.

When populating an HSM system with archival files, that is, files that you want to move quickly to staging media, you can set the *convenient* property (**-C**) on the directory that contains the files. Then, any file loaded into that directory (or subsequently created subdirectories) will be staged out during the next periodic staging run. See the **emchmod** man page for details about the convenient property.

For details on moving data from one EDM to another, see "Moving and Copying Files Between HSM Systems" on page 13-14.

Disabling Filesystem Staging

There are two ways to disable staging. You can temporarily disable periodic staging for an entire system, for a staging template, or for an individual filesystem. Or, you can permanently disable both periodic and demand staging.

Temporarily Disabling Periodic Staging

By changing the enable value from Y to N in either the **emsysconf** command, the **emstconf** command, or the **emfsconf** command, you can temporarily disable periodic staging for an entire system, for a staging template, or for an individual filesystem.

```
emc# emsysconf N 2OR
emc# emstconf CAD N - - - - - - -OR
emc# emfsconf  /mech N - - - - CADOR
```

See the man pages for details regarding command syntax.

The enable value only affects periodic staging; it has no effect on demand staging (crossing a high watermark), user-specified staging (**emstage** and **restage**), baseline backup, or stage-in.

Permanently Disabling Periodic and Demand Staging

Use the following procedure to permanently disable filesystem staging, that is, periodic and demand stage-out and stage-in. This procedure not only disables staging, but it also stages in all staged-out files. If the filesystem does not have the room, you can move the staged-out files to another filesystem. See "Copying Files from One Filesystem to Another" on page 13-14.

To preserve data integrity, you must follow this procedure exactly as described.

1. Make sure you have a valid set of backups for the filesystem.
2. Become superuser.
3. Change permissions on the root of the target filesystem to allow only *rwx* access by root. (Before you change the permissions, do an **ls -lad** to determine what the current permissions are. You reinstate these permissions at a later date.)

   ```
   emc# ls -lad /alpha
   drwxr-xr-x 9 root 512 Sep 4 15:40 /alpha

   emc# chmod 700 /alpha
   ```

4. Disable periodic and demand staging. Note that this step prevents periodic and demand staging, but it does *not* prevent users from staging specific files. At this point, users can still access staged out files.

   ```
   emc# emfsconf -r /alpha
   ```

5. Disable compaction by commenting out or editing the appropriate line in root's crontab file. The default compaction line is as follows:

   ```
   00 1 * * * /usr/epoch/bin/emcompact -c
   >/dev/null 2>&1
   ```

   In addition, abort any compactions in progress.

6. Disable baseline backups from the Backup Configuration interface.

   c. From the EDM main view's Backup menu, select Configure.

   d. On the Server tab select **Disable baseline backups.**

   e. Select Save, then exit EDM.

7. Shutdown the system and then reboot. This ensures that no one is logged in and severs any remote connections to the system.

   Note: Use only /usr/sbin/shutdown -y -i6 -g0
   on an HSM system. Do not use halt or reboot.

8. Use **emfsreport** to determine whether enough free space is available to stage in all of the staged-out files. The important line to look at in the **emfsreport** display is the Virtual GB of data (239.35 MB, in this example):

```
emc# emfsreport /alpha

/alpha                  |  TOTAL    | REGULAR  |   DIR    | SPECIAL  |  SY
========================+===========+==========+==========+==========+=====
Number of files         |    16701 |     5799 |     1818 |        0 |
GB of stagable data     |  0.20034 |  0.20034 |  0.00000 |  0.00000 |  0.0
GB of not stagable      |  0.01340 |  0.00288 |  0.00185 |  0.00000 |  0.0
GB of data staged       |  0.03130 |  0.03130 |  0.00000 |  0.00000 |  0.0
Virtual GB of data      |  0.23935 |  0.22884 |  0.00185 |  0.00000 |  0.0
Stagable Vir-Phs ratio  |  1.195:1 |  1.142:1 |  0.000:1 |  0.000:1 |  0.0
Actual Vir-Phys ratio   |  1.120:1 |  1.126:1 |  1.000:1 |  0.000:1 |  1.0
GB available for working set:   0.230
(About 572.1 to 739.6 days worth.)
(Range given because 1% of virtual space
```

9. If you do not have enough free space to stage in all of the staged-out files, refer to the following section.

10. Run **emfsdeconfig** to complete the procedure.

    ```
    emc# emfsdeconfig /alpha
    ```

Moving Staged Out Files to Another Filesystem

If the filesystem does not have enough space to hold all of the staged-out files, proceed as follows:

1. Change to the /alpha directory.

2. Use **ebcp** to copy the files to another filesystem, which may be stageable or non-stageable. The following example copies all of the files in /alpha to /new_alpha.

   ```
   emc# ebcp . /new_alpha
   ```

3. Delete the files in the /alpha filesystem.

4. Unmount the /alpha filesystem to clear data structures used for stageable filesystems.

   ```
   emc# umount /alpha
   ```

5. Recreate the /alpha filesystem.

Compacting Staging Media

In most cases, staging media is compacted automatically via an **emcompact** entry in root's crontab file. With automatic compaction, **emcompact** automatically determines which volumes to compact.

You can also compact staging volumes manually, if you want to compact any additional volumes. Both automatic compaction and manual compaction use the **emcompact** command.

If you need to compact some staging volumes manually:

1. Use **dbreport**'s compaction report to decide which volumes to compact.

   ```
   emc# dbreport compaction
   ```

   The compaction report is divided into three sections. The most likely volumes to compact are those that are listed in the last few lines of the first section. These are the volumes with the highest percentage of stale files.

2. Use **emcompact** to compact the volumes. In the case of EOs, which have two sides, you need to specify each side, or volume, separately. You can specify volumes by:

   - volume id
   - sequence number (for single-sided media)
   - sequence number and side (for double-sided media)
   - barcode (for single-sided media)

   The following example compacts both sides of disk #10:

   ```
   emc# emcompact EO 10-1 10-2
   ```

   You can type the command without any arguments to find out the legal media types. (In the case of tapes, you can specify a barcode.)

   You can override HSM's file residence policy by running **emcompact** with the **–p** (policy) option. The **-p** option ensures that all files from the compaction source volume are staged out to the compaction output volume and none remain on magnetic disks.

Administering Compaction

There are several things you can do on a regular basis to make sure that compaction is working smoothly.

CAUTION: In order to ensure a complete file restore process, you should disable automatic compaction and **emvck** as soon as you realize that you've lost a filesystem or a significant portion of one.

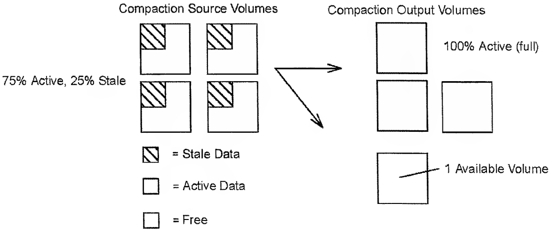
- Check the Media Requests window every morning to see if automatic compaction has blocked.
- Keep a supply of blank, easily accessible and unlabeled volumes, which you can label and allocate as compaction output volumes.

  To increase the likelihood of maintaining a supply of free volumes, you can also convert unrestricted staging trails to restricted ones, or prelabel volumes for a specific trail.
- Check the message files every day.

  If automatic compaction frequently fails to reach the free goal, the system might have reached the limit of its data storage, given the number of volumes in the library unit or the time available for compaction.

  In such a case, use the **dbreport appl_usage** report to determine whether there is enough stale space to reclaim, or whether you should add more volumes to your library unit. For example, in an EO system with volumes averaging 25% stale data, you must compact four volumes in order to get a single free volume (see Figure 13-2). If this rate is unacceptable in your environment, add more blank volumes. Refer to *EDM Server Error Messages* for further details on what to do if automatic compaction fails.

Figure 13-2 **Freeing Volumes for Use**



Compacting Baseline Media

Compacting baseline volumes is similar to compacting staging volumes, but you can only compact baseline volumes manually. Compacting baseline volumes causes all currently active data associated with the volume to be copied to the current compaction volume associated with the baseline trail. New baseline compaction volumes are allocated as necessary.

As a general goal, you should limit the number of active baseline volumes to the number of active staging volumes. If possible, you should also limit the number of active baseline volumes to the number of slots in your library units. This facilitates recovery in the event of a site disaster by ensuring that all of the baseline media (to which active files are attached) will fit in your library units.

To compact baseline volumes:

1. Run **dbreport baseline** weekly and refer to the "pct_stal" column for baseline volumes.

```
emc# dbreport baseline
```

2. Use **emcompact** to compact the *N* most stale optical disks or tapes. (In the case of optical disks, there are two volumes per disk.) The value of *N* varies, but it is at least the number of active baseline media minus the number of active staging media or slots in your library units, whichever is less.

   Therefore, if you have 55 active baseline media, 45 active staging media, and a 50-slot library unit, compact at minimum the ten most stale baseline media.

   This results in a set of compacted baseline volumes. Note that these volumes are not deallocated and available for use until all existing baseline-relative backups that reference them expire.

Clearing Incomplete Bitfiles

As part of nightly maintenance, the system runs the **emscheck** program via root's crontab to clear the incomplete bitfiles that accumulated in the client stores as a result of an interruption of service during a stage-out.

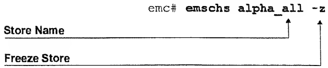
For example, the following line in root's crontab runs **emscheck** every day at 12:30 a.m.:

```
30 0 * * *  /usr/epoch/bin/emscheck >/dev/null 2>&1
```

Moving a Store to Another EDM Server

To move client stores from one EDM server to another, do the following:

1. Use the **emschs** command to freeze the client store.



```
emc# emschs alpha_all -z
```

2. Use the **ebcp** command to relocate the entire contents of the store. Note that you must copy both the magnetic and staged out data. The following example copies all of the

files, including the files that are staged out, in the /alpha_all store on a system named "emc" to the /alpha_new store on a system named "emc2."

```
emc# cd /stores/alpha_all
emc# ebcp -o . | rsh emc2 /usr/epoch/bin/ebcp
 -i /stores/alpha_new
```

3. On the new server, emc2, use **emsmks** to add the store to the server configuration.

```
emc2# emsmks alpha_new -p /stores/alpha_new -c  alpha_new

Directory /stores/alpha_new  already exists, reuse [ y/n] ? y
Store configuration data already exists, reusing.
emsd is running, restarted..[ OK]
```

4. Unfreeze the store on the new server:

```
emc2# emschs alpha_new -w
```

5. Notify the client system of the store's new location. Note that in this case trail_1 is the name of the previous staging trail.

```
alpha# emstconf trail_1 - - - - - - emc2:alpha_new
```

6. Add the new migration tag on the new server:

```
emc2# emschs alpha_new -t new_migration_tag
```

7. Remove the old store from the original server:

```
emc# emsrms alpha_all
```

Gathering Migration Store Statistics

Use the **emsstat** command to display network migration server activity levels. The **emsstat** command gets its information by accessing statistics that are kept in a shared memory segment used by all active EDM Migration server daemon (emsd) processes, or from a statistics file if emsd is not running.

If you type **emsstat** without any arguments, it displays statistics representing server activity since the statistics were last cleared. If you use the **-i** option, **emsstat** displays the ongoing server activity, and, by default, updates the screen in 10 second intervals.

To reinitialize the statistics database enter the following commands:

```
emc# emshalt
emc# rm /usr/epoch/etc/mal/emsd_stats
emc# emsstart
```

See the **emsstat** man page for further details.

Checking a Network Client's Staging Configuration

Use the **emcheck** command to check the EDM Migration client configuration. You must be superuser to run **emcheck**. If you type the command without any arguments it checks the configuration database, warns you of potential problems and corrects inconsistencies. See "Checking the Staging Configuration" on page 13-12 for more information.

Troubleshooting HSM

HSM depends on the interaction of several daemon processes. Most HSM problems are caused by the failure of one of these processes. The following checklist enumerates the steps to take to troubleshoot HSM problems. (Use the **emlistd** command to check whether any of the following processes are running.)

❑ Verify that the HSM file monitor daemons (**emfmd**) are running.

❑ If you are unable to stage in files, verify that the stage-in daemons (**emsid**) are running.

❑ If demand or periodic staging fails, verify that at least one master daemon (**emmasterd**) is running.

❑ If the user-level commands (**emstage**, **embsi**, **emchmod**, and **emls**) are failing, verify that the HSM RPC daemon (**emrpcd**) is running.

❑ Look in the /var/adm/epoch/detail log for error message information.

Restoring a Lost or Damaged Staging Volume

The procedures for restoring a lost or damaged staging volume vary, depending on whether or not baseline backup is installed.

Restoring a Lost or Damaged Staging Volume (No Baseline Backup)

If baseline backup is *not* enabled:

1. Use the **dbreport volumes** report to find the volids of the missing volume(s). (The volid is a 16-digit hexadecimal number.) For two-sided media, you need to get the volids of both sides.
2. If the lost volume happens to be the current staging volume, the current compaction volume, or an active backup volume, you need to use **em_new_volume** as follows:

```
emc# em_new_volume staging_trail_name
```

3. Locate the files staged to the missing volumes by using **emfind**(1):

```
emc# emfind / \( -staged_to 0000111122223333 -o\ -staged_to 0000111122223334 \) -
print > /usr/tmp/files
```

Note: If **emfind** encounters a pathname that is too long, it generates an error message. The pathname is not added to /usr/tmp/files, and the subsequent restore are incomplete. Some manual intervention (descending into directories and rerunning **emfind**) is necessary in such cases.

4. Delete the missing volumes from the database using **evmrmvol**:

```
emc# /usr/epoch/bin/evmrmvol -v 0000111122223333

emc# /usr/epoch/bin/evmrmvol -v 0000111122223334
```

5. Restore necessary files:

```
emc# ebrestore -D server -c server -w workitem -d -f /usr/tmp/files
```

Restoring a Lost or Damaged Staging Volume (Baseline Backup is Enabled)

If baseline backup *is* enabled, you can use **ebcheck** to restore a staging volume. The steps are as follows:

1. Using **dbreport volumes**, obtain the IDs of the missing volume(s). For two-sided media, you need to get the IDs of both sides.
2. Delete the missing volumes from the database using **evmrmvol**:

```
emc# /usr/epoch/bin/evmrmvol -v 0000111122223333

emc# /usr/epoch/bin/evmrmvol -v 0000111122223334
```

3. If the missing volumes are baseline backup volumes, simply invalidate the pointers to the baseline volumes:
   ```
   emc# ebcheck -a -i
   ```
4. Otherwise, if the missing volumes are primary staging trail volumes, restore as many of the files as possible from the baselines:
   ```
   emc# ebcheck -a -i -b1 -b2
   ```
5. Restore any remaining files from the full and incremental backup volumes:
   ```
   emc# ebcheck -a -r1 -r2 > /usr/tmp/files
   ```
6. If /usr/tmp/files is non-empty:

```
emc# ebrestore -D server -c server -w workitem -d -f /usr/tmp/files
```

Restoring a Lost or Damaged Staging Trail

This method uses baseline volumes as a temporary staging trail, which lets you bring the system back up much sooner. The procedure is very quick, and you can generate your new primary staging trail while the system is up and providing service:

1. Delete each missing volume from the database as before.
2. Make the baseline backup volumes also act as a copy of the primary staging trail:
   ```
   emc# ebcheck -a -i -c1 -c2
   ```

3. Restore any remaining files from the full and incremental backup volumes:

   ```
   emc# ebcheck -a -r1 -r2 > /usr/tmp/files
   ```

4. If /usr/tmp/files is non-empty:

   ```
   emc# ebrestore
   ```

5. Using **restage**, generate a new primary staging trail for each filesystem as appropriate:

```
emc# restage -t trailname / \( -fstype nfs -prune \) \ -o -staged_to_trail
baseline_trailname
```

Restoring a Lost or Damaged Filesystem

These steps restore a filesystem to its state as of its last backup, after a complete loss due to disk failure and the like. The steps assume the staging trails, backup catalogs, and the root filesystem all are still intact. The steps are designed to avoid all accesses to the filesystem while it is being restored, whether local or remote, so that users do not see a filesystem that is only partially recovered.

1. Create a new empty filesystem, the same size as the original one or larger:

   ```
   # mkfs -F vxfs -o inosize=512 /dev/rdsk/cXtYdZ
   ```

2. Mount the filesystem to a temporary location (called /tmp_doc in the examples below):

   ```
   # mkdir /tmp_doc
   # mount -F vxfs /dev/cXtYdZ /tmp_doc
   ```

3. If the filesystem was configured for HSM, you must now configure the new filesystem for HSM. This step must be done before recovering any data, or staged out files are not recovered correctly.

   Use the EDM HSM configuration interface, just as if you were configuring a filesystem for the first time. The filesystem may be assigned to the same staging trail as the original, or to a different staging trail; it makes no difference.

4. Using the EDM Restore interface, restore the last backup of the filesystem you lost (called doc here) onto the new filesystem (mounted here at /tmp_doc).

   Mark everything in the filesystem except for the .-EPOCH-. and lost+found directories.

5. Change to the /tmp_doc/doc directory, and move everything to /tmp_doc.

   ```
   # cd /tmp_doc/doc
   # mv * /tmp_doc
   # cd ..
   # rmdir doc
   ```

   Be sure to check for files or directories whose names begin with "." since they are not moved by the mv command. The **rmdir** command fails with "Directory not empty" in such cases. The **ls -a** command lists these files; they have to be moved manually

6. If the new filesystem (/tmp_doc) was configured for HSM, remove it now from the staging configuration database. There is no need for this configuration now, since the original filesystem (/doc) should still be in the staging configuration database.

7. Umount the filesystem from its temporary mount point, and remount it under its real name.

   This example assumes that the same physical device address is being used for the new /doc (the usual case if a new disk was brought in to replace a failed one). If the device address is now different, be sure to update the /etc/vfstab entry for the filesystem.

   ```
   # cd /
   # umount /tmp_doc
   # rmdir /tmp_doc
   # mount /doc
   ```

Part III
Logs and Reports

14 Start of Backup and Related Processing

The EDM Backup software automatically initiates backups, processes catalogs, and generates backup reports. You can also initiate these functions manually, or edit the crontab file to change how these functions occur automatically. This chapter contains the following topics:

- Backup Processing
- Catalog Processing

Backup Processing

Following are methods of initiating backup processing with EDM Backup:

- by using the Backup Activity Wizard
- automatically from crontab (see page 14-3)
- manually from command line (see page 14-7)

The Backup Activity Wizard enables you to start new, queued, or failed backups, stop running backups, or manage the backup queue. You access this wizard from the Main window of the EDM GUI.

> **Note:** You must have root privileges or be an EDM Backup Administrator to use the Backup Activity Wizard.

For automatic processing, EDM Backup uses the crontab facility to issue the **ebbackup** command at a particular time each night to start backups. The command specifies a backup schedule template, which contains scheduling parameters. You can create additional crontab entries (using the Backup Configuration Wizard, by editing the file, or from the Backup Configuration window of the EDM GUI) for any schedule templates that you add. You can optionally specify a particular work group or work item, backup level, or specific day with the **ebbackup** command, but you would typically use the Backup Configuration Wizard or the backup template.

To start network backups manually, you can issue the **ebbackup** command from the command line. Again, you can choose to specify a backup schedule template, but you would more typically specify a particular work group or work item, backup level, or specific day when starting backups from the command line.

To start Symmetrix Connect backups manually after successful client configuration, issue the **eb_dc_backup** *work_item _base_name* command. Refer to the *EMC Data Manager Symmetrix Connect User Guide* for more information on starting these types of backups and related processes.

> **Note:** However you initiate backups, make sure your clients are configured and that you have labeled your media and inserted that media into the appropriate library unit.

For additional information, refer to "Scheduling" on page 3-4.

The following sections describe starting backups in the crontab file or manually from the command line.

Backup Activity Wizard

As mentioned above, the Backup Activity Wizard enables you to start new, queued, or failed backups, stop running backups, or manage the backup queue. You access this wizard from the Main window of the EDM GUI.

Click this icon in the Main Window to start the Backup Activity Wizard.

In the wizard panels, you select a backup operation, select the objects on which you want to operate, choose backup options, and confirm your actions. You can then monitor the progress of the backup operation that you initiated in the Main window.

Refer to EDM online help for more information.

Automatic Nightly Processing

You can use one of the following methods to configure the crontab facility to schedule automatic backups for each defined schedule template in the configuration file:

- Backup Configuration Wizard (which you access from the EDM Main window toolbar)

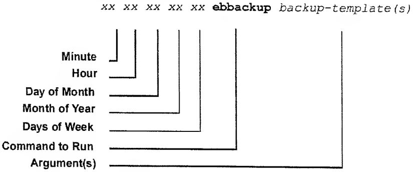
- Backup Configuration window (refer to EDM online help for more information)
- CLI

By default, the backup schedule template runs automatically at 6:00 p.m. every night.

Note: An error message appears if scheduling a backup via cron is unsuccessful.

Each line in root's crontab file has several fields of information. Figure 14-1 shows the format of the crontab entry that invokes the **ebbackup** program.

Figure 14-1 **Root's Crontab File Entries**



```
xx xx xx xx xx ebbackup backup-template(s)
```

You can edit root's crontab file (/var/spool/cron/crontabs/root) to change when to begin backups. If you add any backup schedule templates, you edit root's crontab file to specify when to begin backups.

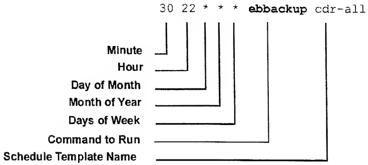
You can also schedule a backup in the crontab file within the Backup Configuration window of the EDM GUI. In the window, you select a work group for backup and the time that the backup is to occur. You can also indicate whether you want to schedule a failed backup, or use new media for a backup. (For more information, refer to Chapter 2 of this manual, EDM online help, and the **ebbackup** man page.)

After you edit the file, the backups occur automatically. At the specified time each day, root's crontab file invokes the **ebbackup** command, which starts overall backup processing. Edit the file again only when you want to change the nightly backup run time.

Figure 14-2 shows a sample crontab entry, which starts a backup using the schedule template named "cdr-all."

The template "cdr-all" specifies the work groups to back up and the trailset to write the backups to. Asterisks(*) represent unspecified fields.

Figure 14-2 **A Sample Crontab File Entry**



```
30 22 * * * ebbackup cdr-all
```

The sample crontab entry in Figure 14-2 indicates the following:

- the Minute field specifies to start the backup 30 minutes after the hour.
- the Hour field specifies the hour at which to start the backup. In this example, the backup will start at 10:30 p.m. The length of time **ebbackup** runs depends on the length of the shift as defined in the backup template or when all scheduled backups finish—whichever occurs first.
- the Day of Month field indicates all days of the month (*).
- the Month of Year field indicates all months (*).
- the Days of Week field indicates all days of the week (*).

- the Command to Run field specifies the command (**ebbackup**).
- the Schedule Template Name field specifies the backup schedule template on which to run the backup (cdr-all).

Automatic Scheduling

The **ebbackup** command starts backups for a backup schedule template. By default, a backup schedule template specifies automatic scheduling of backups for all of the work items affected by it. When EDM Backup starts it uses the rotation period, the rotation scheme, and the backup shifts that are specified in the named schedule template to determine what work items to back up during that backup session.

EDM Backup manages the backups, performing one full backup (level 0) for each work item each rotation period, and scheduling level 9 backups on the remaining days in the rotation period.

EDM Backup's ability to automatically balance the backup work load frees you from the task of manually assigning each client to a backup schedule. Automatic scheduling also adjusts to clients that are down during the scheduled backup.

Custom Scheduling

You can also specify custom scheduling for the backup. In custom scheduling, the backup schedule template explicitly defines the work items and the days on which they are scheduled for backup. You can specify the custom schedule for the template within the Backup Configuration Wizard or the Backup Configuration window of the EDM GUI.

Backups that you schedule by using the custom schedule directives in the backup template are initiated using root's crontab file to begin processing for the template, just as for automatic scheduling.

Note: If a client is unavailable on its scheduled backup day, the backup does not automatically reschedule the backup to another day as auto scheduling does.

It is also possible to use the **ebbackup** command to directly specify particular backup levels on certain days for certain work groups or work items. This alternative command format is described in the next section.

Command Line Processing

You can choose to initiate all backups by using the **cron** facility or command line.

The **ebbackup** command enables you to specify the level of backup, work group, work item, priority, and/or trailset to use for a backup. You can use the **ebbackup** switches in combination with the crontab facility to schedule a command for a certain time each day to schedule each backup.

The **ebbackup** command options are useful when you want to schedule a specific backup. For example, you may want to force an immediate full (level 0) backup for a particular client's work item before performing an operating system upgrade on that client. You can also use the command line to schedule or reschedule backups that have failed. Refer to the man pages for **ebbackup** for details about the available options.

When you specify a backup from the command line, this backup overrules any backups scheduled by automatic or custom scheduling.

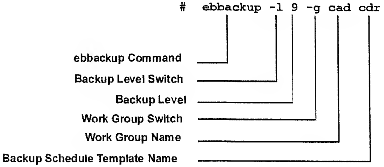
Note: To run a command line backup for a particular backup level, you must first define that level in a trailset.

The following examples show how to use the command line to backup a work group, a work item, and an HSM work item. For more information see the **ebbackup** man page.

The command line in Figure 14-3 starts level 9 backups for the clients in the work group named "cad."

Figure 14-3

Backing Up a Work Group

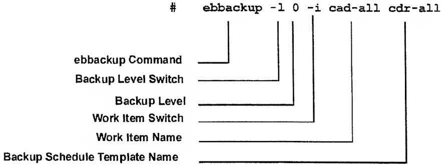


```
#  ebbackup -l 9 -g cad cdr
```

The command line in Figure 14-4 starts level 0 backups for the work item named "cad-all."

Figure 14-4

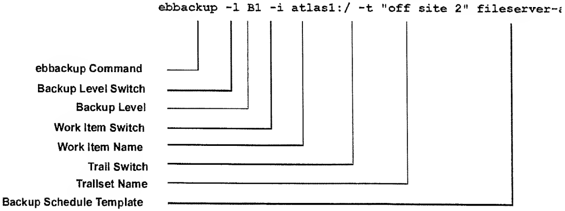
Backup up a Work Item



```
#  ebbackup -l 0 -i cad-all cdr-all
```

The command line in Figure 14-5 starts baseline backups for the HSM work item "atlas1:/" to the "off site 2" trailset.

Figure 14-5 **Backing Up an HSM Work Item to a Trailset**



```
ebbackup -l B1 -i atlas1:/ -t "off site 2" fileserver-
```

Catalog Processing

When a backup completes, the raw data for the associated catalog exists on the EDM Backup server. The catalog daemon (**ebcatalogd**) must process the raw catalog before the restore process in the EDM Restore window or the command **ebrestore** can use it. You can have catalog processing performed concurrently with backups, or you can schedule catalog processing for a later time so that this task does not slow down client backups.

Normally, the /etc/rc3.d/s30ebs script starts **ebcatalogd** at boot time.

To exercise greater control over the catalog processing schedule, you can start and stop **ebcatalogd** manually, or you can control it automatically via root's crontab file. To start catalog processing, run **ebcatalogd** without arguments. The catalog daemon places itself in the background when it runs, and terminates if another catalog daemon is already running. To stop catalog processing, run **ebcatalogd** with the **-halt** option.

Figure 14-6 shows a crontab entry that starts catalog processing at 6:30 AM each day, which is after the site's backup shift ends.

Figure 14-6 **Starting Catalog Processing from Crontab**

```
30 06 * * * /usr/epoch/EB/config/daemon_startup -ebcatalogd
```

Figure 14-7 shows a crontab entry that stops catalog processing an hour before the backup shift begins again at 10:30 p.m., the site specifies halting catalog processing at 9:30 p.m.

Figure 14-7 **Halting Catalog Processing from Crontab**

```
30 21 * * * ebcatalogd -halt
```

15 Message Logging

The EDM Backup and HSM software maintains a message logging system that uses both the system log daemon, **syslogd**, and circular logs to record significant events. These messages can be written to log files or to the system console. The configuration file (/usr/syslog.conf) determines the error conditions that are logged and where the messages are sent.

This chapter contains the following sections:

- Message Logging Features
- Syslog Message Files
- Circular Log Files
- Log Message Format
- Default syslog Configuration File

Message Logging Features

The message logging system offers the following features:

- Automatically creates log files:

  During system installation several message log files are created automatically. These files are specified in the /etc/syslog.conf file, and described in "Default syslog Configuration File" on page 15-6.

- Uses **cron** to mail messages to system administrators:

  For sites with a mail facility, **cron** starts a script that mails log messages which describe the system activities (anomalies only) for the previous 24-hour period to the system administrator. (See "Running Procedures Automatically via Cron" on page 2-6.)

- Groups files into logical categories:

  The **syslogd** sends log messages to specific log files depending on the type of message. Because EDM Backup software groups messages into separate log files, you can choose to look at the log file that is most appropriate for the task at hand. For example, all messages that describe maintenance activities are sent to one particular log file, while messages that show error audit trails are sent to another.

- Monitors interaction of EDM subsystems:

  Subsystem messages work together to give a complete view of system activity. This means that when an event occurs, all affected subsystems log a message. By looking at messages that are sent from different subsystems, you can see the relationship of each subsystem's activity.

Syslog Message Files

All syslog messages are written to log files that reside in /var/adm/epoch.

Note: All of the log files, except for the daily log, are rotated or archived in /usr/epoch/adm, where they remain for four weeks.

The messages in these files enable you to determine the cause of a system problem. The log files are named the following:

- concise
- daily
- debug
- detail
- mntfault
- lu_hardware

Table 15-1 explains when to look at each message file:

Table 15-1 **When to Look at the Syslog Message Files**

If you want to:	Look at this file:
determine quickly if any system problems have occurred. If the file is empty, you know the system is operating properly. You should view this file daily.	/var/adm/epoch/concise
see system problems for the past 24 hours. The daily log is a subset of the concise log.	/var/adm/epoch/daily
check the debug log. Debugging must be turned on according to the directions in the /etc/syslog.conf file. This file is primarily for use by Customer Service personnel.	/var/adm/epoch/debug
see additional information about the errors that appear in the concise log.	/var/adm/epoch/detail
view a list of volume requests (which require operator assistance) and the audit trail for volume allocation and erasure. You can forward these messages to other systems and/or to the system console for monitoring.	/var/adm/epoch/mntfault
see additional information about hardware errors that appear in the detail log	/var/adm/epoch/lu_hardware

Circular Log Files

You can configure message logging to bypass the standard system logs and write messages to circular logs. Circular logs are located in a central directory or in application-specific directories. The following directories contain circular logs:

- /usr/epoch/adm/circular
- /usr/epoch/etc/*subdirectory*

The names of the circular log files that are located in the central directory /usr/epoch/adm/circular are based on the daemon name. For example, the name of the circular log for the volume management erase daemon is vmerd.log.

Volume management maintains its circular log in /usr/epoch/etc/vm. This provides the system administrator with access to all VM-related files in one directory. For this same reason, circular logs for each Library Manager reside in individual subdirectories of /usr/epoch/etc/lm.

Circular logs that reside in VM and LM directories are named clog (for circular log).

You can use the **fuser** command to show all processes that have a given file open. For example:

```
#   fuser -f /usr/epoch/etc/lm/hp_mf_sa/clog
```

Log File Rotation and Archival

If you are using the migration option, the concise and mount fault logs are rotated in /usr/epoch/adm.

The detail and debug logs are archived in /usr/epoch/adm for one year in systems on which EDM Migration is installed. The logs are archived in *year-month* subdirectories as shown in the following examples:

```
/usr/epoch/adm/1999-10

/usr/epoch/adm/1999-11

/usr/epoch/adm/1999-12
```

Within each subdirectory, the archived logs are written to a detail.*day* file. For example, the following file contains the detail log for December 23, 1999.

```
/usr/epoch/adm/1999-12/detail.23
```

If more than one rotation occurs on a single day, another suffix is added as shown in the following examples:

```
/usr/epoch/adm/1999-12/detail.23.0
```

```
/usr/epoch/adm/1999-12/detail.23.1
```

```
/usr/epoch/adm/1999-12/detail.23.2
```

The highest number suffix (detail.23.2 in this example) represents the most recent log.

Log Message Format

Each log message provides the following information:

- date/time string
- host name that identifies the name of the system on which the event occurred
- name that identifies the process that generated the message.
- optional user ID (usually supplied if an interactive program is logging the message). Many subsystems can only be run by root and therefore omit this field.
- process ID number that appears between square brackets. (Kernel messages, which are identified by the prefix *vmunix*, do not have this field.)
- optional layer name, sometimes prefixed by a subsystem name, that provides EMC customer service with additional information
- message number that uniquely identifies the message. Using the prefix that immediately precedes the pound sign (#). This prefix is either the process or layer name, depending on the message.

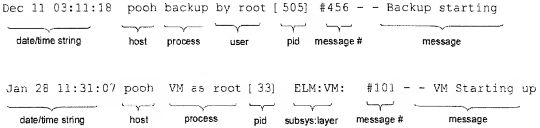
- brief free-form description of the condition.

To summarize, each log message conforms to the following syntax:

```
date/time host process <as user> pid <<subsystem:> layer:> message # -- message
```

Figure 15-1 lists some sample log messages:

Figure 15-1 **Sample Log Message**



```
Dec 11 03:11:18  pooh backup by root [ 505]  #456 - - Backup starting

Jan 28 11:31:07 pooh  VM as root [ 33]  ELM:VM:  #101 - - VM Starting up
```

Default syslog Configuration File

A default, sample /etc/syslog.conf file is provided when you install the software. It contains the following lines in which the first column specifies the error condition and the second column specifies the file to which the error is logged.

```
kern.err;local5.err    /var/adm/epoch/concise
kern.err;local5.err    /var/adm/epoch/daily
local5.warning         /var/adm/epoch/mntfault
kern.info;local5.info /var/adm/epoch/detail
* .debug                  /var/adm/epoch/debug
```

Note: The concise and daily logs receive the same messages. The messages in the daily log are truncated each day.

16 Backup Reports and Log Files

When you run a backup or restore, EDM Backup saves information that you can then access through log files or reports. You can review this information to verify that your applications are executing correctly.

Reports are available online in the Backup Report window of the EDM GUI. You can execute reports on successful, failed, active, and queued backups on your local EDM and on multiple EDMs which are set up in a domain. This window enables you to create, modify, and print backup reports to look at key areas, such as performance within specified time periods, work items with poor performance, or failed work items. (For more information see the online help for the EDM Backup Report window.)

Some reports are generated automatically. You can run the **ebreport** commands for other reports from the command line or insert them into your crontab file for automatic processing.

This chapter describes the following reports and logs (for more information, refer to the appropriate man page).

- Report and Log Usage
- Executing Reports from the EDM GUI
- Report and Log Summaries
- Backup Reports
- Backup Media Reports
- Backup Duplicate Reports
- Backup History Reports
- Backup Disaster Reports
- Backup Baseline Reports
- Backup Completion Reports
- Backup Failure Reports
- Backup Coverage Reports
- Volume Reports
- Log Files

Report and Log Usage

You receive logs automatically, whereas you need to request most reports. After a backup or restore finishes, you receive an e-mail message that indicates whether the operation was successful. (You can also monitor backups in progress and then generate reports via the EDM graphical user interface (GUI); refer to "Executing Reports from the EDM GUI" on page 16-3).

If the backup or restore is successful, you do not have to review any of the logs and reports. However, these reports include information about your system that you may want to know or monitor.

If your backup fails, you receive an email message that tells you the backup failed. You should then look at the backups.log file, which contains clues as to why the backup failed. If you see a message such as "Client not available," a client may have gone down during the backup or your system may have a network problem. You should review the backups.log file on the client that failed to see what information that file has on the backup.

A MINIMAL disaster report is generated automatically each time the LOCAL_DATABASE work item completes. You need the information in this report to perform a disaster recovery. The disaster report, by default, is mailed to the backup administrators, printed, and saved to disk.

Note: Refer to Chapter 19 "Being Prepared for a System Disaster".

Regardless of whether your backup succeeded or failed, you may want to review the backup coverage reports. Each report displays information about the filesystems that were not backed up. It displays the client names, the size of each filesystem, the name of each filesystem, and a summary line with grand totals for each column of information.

Executing Reports from the EDM GUI

You can execute reports in the EDM graphical user interface (GUI). These reports can be reports for a local EDM or domain reports for a group of EDMs. The online help for the Backup Report window explains how to set up and use a domain, and lists the limitations of domain reporting (for example, that a domain cannot span a firewall or reconcile time differences on machines).

Objects in the Main window such as the EDM server, a client, or a work item, are colored to designate a successful, failed, or queued backup.

You can configure, run, and print backup reports on specific areas such as failed work items or work items with poor performance.

Click on this icon in the Main window toolbar to access the backup report module.

For more information about active backup reporting, refer to EDM online help, "Backup Report Overview."

Report and Log Summaries

You can initiate reports manually by using the following commands, or you can add them to your crontab file for automatic generation. For more information on a specific report, refer to the man page for the report.

Table 16-1 lists the reports that EDM Backup generates.

Table 16-1 **EDM Backup Reports**

Report Name	Command	Information
Backup	ebreport backup	Contains a summary of the backup activity performed by the server.
Backup media	**ebreport media**	Contains information about the media to which EDM Backup wrote backup data.
Backup duplicate	ebreport duplicate	Contains information about media duplication processes. (Refer to Chapter 9 "Media Duplication".)
Backup history	**ebreport history**	Displays information about the backups performed on the server. Use the command options to display specific details.
Backup disaster	**ebreport disaster**	Contains a combination of other reports including a list of media for each backup trail, a detailed record of client backups, copies of the key configuration-file settings, and a history.
Backup baseline	ebreport baseline	On HSM systems, contains a summary of baseline backup activity as reflected in the saveset database.
Backup completion	none [1]	Confirms backup operations.
Backup failure	none 1	Notifies you if your backup fails.
Backup coverage	**ebreport coverage**	Displays information about which filesystems were backed up and which filesystems EDM Backup is not backing up.

[1] There is no command available since the report is generated automatically.

For detailed descriptions, see the following sections in this chapter.

Note: The examples in this chapter are from different servers and different configurations and times. They cannot be compared to each other.

Table 16-2 lists the log files that are generated by EDM Backup and located in /usr/epoch/EB/log on the server. For detailed descriptions, see "Log Files" on page 16-35.

Table 16-2 **EDM Backup Log Files**

Report Name	Information
backups.log file	Displays an audit trail of backup-related activities on the server and the client.
recoveries.log file	Displays ebrestore operations on the server and the client.
ebcat.log file	Contains catalog processing startup and shutdown times.
Server log file (template level)	Displays backup information about activities for a single backup schedule (template).

Backup Reports

The **ebreport backup** command presents a summary of EDM Backup activity. It reports the status of the *most recent* backup run (unless the **-recent** or **-since** option is given). The work items are grouped by template (backup schedule) name, and trailset (media set). For a given work item name, the most recent backup is shown first.

Run **ebreport backup** every day to verify that all of the scheduled backups completed.

Table 16-3 **ebreport backup Command Information**

Option	Argument Definition	Description
-client *clientname*	*clientname* is the client whose backup history you want to display.	Displays only the specified client's work items that are backed up to the backup server. When the client name used is *servername*, it displays backups of server filesystems and databases, but does not display backups of filesystems or databases on remote clients.
-level *levelnumber*	*levelnumber* is a level from 0 to 9, a range of levels (for example, 0-8), or B1 or B2, for which you want to display backup history.	Displays only the backups of the specified levels (unless you use the other options to limit the report coverage).
-recent		Displays all of the work items that were backed up by EDM Backup, from the second most recent level 0 backup to the present.
-since *date* [*time*] **-until** *date*	*date* is the date in the format *mm/dd/yy*. *time* (optional) is the time in the format [*hh*:*mm*[:*ss*]].	Limits the backup report to a range of dates. Use **-since** to show the backups that occurred on or after a particular date or use -**until** to display the backup history that occurred on or up to a particular date.

Table 16-3 **ebreport backup Command Information (Continued)**

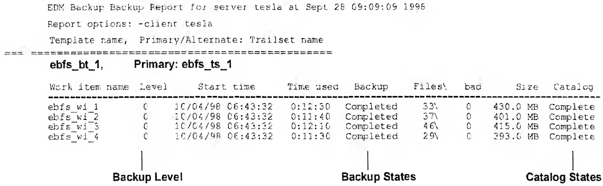
Option	Argument Definition	Description
-trailset primary | alternate		Displays only the work items that were backed up by a primary or alternate set of media (trailset).
-template *name*	*name* is the template (schedule) from which backups are selected. If "*" is used for *name* all templates are selected (the default). Note that "* " must be quoted on the command line	Selects only the backups that were created for the named template.
-workitem *name* or **-item** *name*	*name* is the named work item for which backups were created. If "*" is used for *name* all work items are selected (the default). Note that "* " must be quoted on the command line.	Selects only the backups that were created for the named work item.

You might see database work items with the same name except for an added suffix in the form "`:stripe_`*n*`_of_`*m*`.`" This occurs if the backups were *striped*. If the backups were striped, you also see a suffix on the template names. For example:

```
hamster:master+:sybase:stripe_1_of_2.
```

Figure 16-1 shows a sample backup report.

Figure 16-1 **EDM Backup Backup Report**



```
    EDM Backup Backup Report for server tesla at Sept 28 09:09:09 1998
    Report options: -client tesla
     Template name,  Primary/Alternate: Trailset name
=== ============================================
     ebfs_bt_1,      Primary: ebfs_ts_1

     Work item name  Level       Start time       Time used   Backup     Files\   bad       Size   Catalog
    ---------------------------------------------------------------------------------------------------------
    ebfs_wi_1          0      10/04/98 06:43:32   0:12:30   Completed     33\      0     430.0 MB  Complete
    ebfs_wi_2          0      10/04/98 06:43:32   0:11:40   Completed     37\      0     401.0 MB  Complete
    ebfs_wi_3          0      10/04/98 06:43:32   0:12:10   Completed     46\      0     415.0 MB  Complete
    ebfs_wi_4          0      10/04/98 06:43:32   0:11:30   Completed     29\      0     393.0 MB  Complete
```

Table 16-4 describes the different backup states that can appear in the Backup Report.

Table 16-4 **Backup States**

State	Description
Started	Backup is underway (or it was interrupted before finishing)
Partial	Backup was manually shut down before finishing
Incomplete	An error caused the backup to fail
Completed	Backup finished successfully
Unsuccessful	Backup completed with errors
Failed	No files were backed up, possibly due to a misconfigured work item
Timed-out	Client connection timed-out

Table 16-5 describes the different catalog states that can appear in the Backup Report.

Table 16-5 Status of Catalog Processing

State	Description
Partial	Catalog is being created (or it was interrupted before finishing)
Unsorted	Intermediate state in the post processing of the catalog.
Sorted	Intermediate state in the post processing of the catalog.
Complete	Post processing completed
Delta	Catalog was reduced to a collection of changes against the catalog of the subsequent backup
Expired	Online catalog for the backup was deleted

Backup Media Reports

You can use the **ebreport media** command to display a list of all volumes to which EDM Backup wrote backup data. Use one of the options in Table 16-6 to limit the report size.

Table 16-6 **ebreport media Command Information**

Option	Description
-active	Displays the volumes that EDM Backup is currently using to write backup data.
-offsite	Displays the volumes that are marked for offsite storage.
-onsite	Displays the volumes that are marked as onsite (usually after their status was changed from offsite to onsite).
-template *name*	Displays the volumes that the named template uses.
-trailset *name*	Displays the volumes that the named media set (trailset) uses. If **-template** is used, only "Primary" or "Alternate" (both of which are defined in the Backup Configuration window), can be specified.
-trail *name*	Displays the volumes that the named trail uses within a trailset.
-orphans	Displays a list of orphaned volumes. This cannot be combined with any other command line option.
-no_baselines	Prevents the baseline media report from being generated on an EDM with the HSM option. This report is never displayed on a system without HSM.
-help	Displays command line options for **ebreport media**.

A *media rotation* is the collection of volumes that a single backup schedule template writes to a particular trailset (media set) and trail during a single scheduled rotation period.

Volumes are grouped into media rotations. The media report contains one section for each schedule template, trail, and trailset. The report lists within each section the volumes for each media rotation with the date, rotation ID, number of backups, and whether media duplication was used.

Figure 16-2 shows a sample report that **ebreport media** generates:

Figure 16-2 **EDM Backup Media Report**

```
Summary of all media, listed by media rotation groups

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset

09/30/1998 12:54:42 Rotation ID:4CD84987.F6BECF8D.00000200.54028F30, 4 backups

   Media duplication used on 1 copy

*Orig Vol: 60D84A1170094B3E (BNY574), Seq #: 000024 in TLU: at_dlt_3264_0, media: DLT tape

 Dup  Vol: 73D8745B3E0384A5 (BDE133), Seq #: 000028 in TLU: at_dlt_3264_0, media: DLT tape

     Duplication State: Done, Successful, Duplication Date 05/08/1999 16:06:04
```

Descriptions

Section Header

```
Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary
```

Shows Template Name, Trail Name, and Trailset.

Rotation Header

```
09/30/1998 12:54:42 Rotation ID:4CD84987.F6BECF8D.00000200.54028F30, 4 backups
                                        Media duplication used on 1 copy
```

Shows Backup Date and Time, Rotation ID, Number of Backups on Media, Use of Media Duplication.

Volume Entries

```
*Orig Vol: 60D84A1170094B3E (BNY574), Seq #: 000024 in TLU: at_dlt_3264_0, media: DLT tape

 Dup  Vol: 73D8745B3E0384A5 (BDE133), Seq #: 000028 in TLU: at_dlt_3264_0, media: DLT tape
```

Shows Asterisk for most recent volume in the rotation, Blank for all others.
Then Original or Duplicate, Volume ID, (Barcode), Volume Sequence Number, TLU Type, and Media Type.
Duplication information follows with Duplicate Volume ID, (Barcode), Volume Sequence Number, TLU Type, Media Type.
The Duplication State, Duplication Date and Time follow.

Backup Duplicate Reports

The **ebreport duplicate** command displays a list of the original and duplicate volumes that are currently allocated to Epoch-Backup. These volumes are grouped into media rotations. Media rotations in the report are grouped by template, trailset, and trail.

The first line in the report provides the status for the entire rotation; one or more lines follow that show the volumes allocated to the rotation. The rotation status line contains the time that the rotation was created, the rotation ID, the number of partial or complete backups written to the rotation, whether media duplication is enabled for this rotation, and if so, how many duplicates were made. For each original volume the following duplication infonnation appears: duplication state of the volume (Scheduled, Done, etc.), whether the duplication is up to date, duplication status (Empty or Old), and mode of duplication.

Information for each original volume in the report includes the 16-digit volume ID, volume barcode in parentheses, volume sequence number, library unit in which the volume resides, and media type (e.g., DLT or EO). A "*" precedes the last original volume that was allocated to the rotation.

If the original volume has an allocated duplicate volume, the line for each duplicate volume includes the same information as that for the original volume. If a duplicate volume exists, the display may include the total tape padding blocks that were duplicated for the last duplication, duplication start and end times, total duplication time, and duplicate expiration date.

Note: Duplicate volumes that were created before this release of the EDM software may not display all of this information.

Duplicate Command Options

Table 16-7 lists the **ebreport duplicate** command options and related information.

Table 16-7 **ebreport duplicate Command Information**

Option	Argument Definition	Description
-active		Lists only the volumes that were last allocated to each active rotation. Also lists the date and time the trail was last written.
-offsite		Lists only the volumes that are marked for offsite storage.
-since *date*	*date* is the date for which you wish to view duplication history.	Displays only those volumes for which duplications were attempted on or after the specified date.
-template *template name*	*templatename* is the backup template for which you want to display duplication history.	Displays only the volumes that the given template uses. If this option is not supplied or if the given template is "*", all templates appear in the report.
trail *trailname*	*trailname* is the backup trail for which you want to view duplication history.	Lists only the volumes that the given trail uses within a trailset. If this option is not supplied or if the given trail is "*", all trails appear in the report.
-trailset *trailset name*	*trailset name* is the trailset for which you want to display duplication history.	Displays only the volumes that the given trailset uses. If you use **-template**, you can specify only "primary" or "alternate;" otherwise, you can specify any valid trailset name. If this option is not supplied or if the given trailset is "*", all trailsets appear in the report.

Sample Backup Duplicate Report

Figure 16-3 shows a sample report using the **ebreport duplicate** command:

edm# **ebreport duplicate**

Figure 16-3 **EDM Backup Duplicate Report**

```
edm# ebreport duplicate

Rotations for Template "usr_bin", Trail "usr_bin_DLT", Primary Trailset


09/15/1998 09:46:51 Rotation ID:A1D7F9BD.71812B77.00000200.F206F11B, 104
backups
     Media duplication used on 1 copy


Duplication State: Done, Old, Mode: New
*Orig Vol: A1D7F9BD71812B77 (BDE098) Seq. #: 000025 in TLU: at_dlt_3264_0,
media: DLT tape
 Dup Vol: 40D81EE7477F8BDA (BDE146) Seq. #: 000017 in TLU: at_dlt_3264_0, media:
DLT tape
Total Blocks: 349028  Start Time: 09/22/1998 10:54:26  End Time: 09/22/1998
13:06:21
Duration: 001 Hrs. 31 Min., Duplicate Expiration Date: 12/17/1998


09/30/1998 12:57:57 Rotation ID:65D8498A.FD4DC69B.00000200.540819F4, 2 backups
     Media duplication used on 1 copy
```

Backup History Reports

The **ebreport history** command displays reports about the backups that an EDM Backup server and its clients perform. The top line of each report lists the name of the EDM Backup server and the report creation date. Under this line, EDM Backup lists each backup template (schedule) that it backed up, and for each backup template it lists the work items that the template backed up.

For each work item, the report lists the history of backups, one line per backup, with the most recent backups first. The line for each backup includes the time that the backup occurred, the backup level, the backup ID, backup status, the number of files or directories backed up, the backup expiration date, and backup recovery status. (If a backup cannot be recovered, a "NO" appears in the Rcvr field of the history report, which implies that catalog processing needs to be done for that backup.)

If you run the command without any options, the history report can be large. You can use several options to restrict the scope of the report. Use the command options singly or in conjunction with one another to select a restricted set of backups on which to report. The next sections describe the options to the **ebreport history** command.

History Command Options

Table 16-8 lists the **ebreport history** command options and describes the information that is available through the command.

Table 16-8 ebreport history Command Information

Option	Argument Definition	Description
-client *clientname*	*clientname* is the client whose backup history you want to display.	Displays all of a client's work items that are backed up to the EDM Backup server.
-workitem *workitemname* or **-item** *workitemname*	*workitemname* is the client's work item for which you want to display backup history.	Displays a single work item's most recent backup history.
-template *templatename*	*templatename* is the backup template for which you want to display backup history.	Displays all of the client work items that were backed up by the backup template (schedule).
-since *date* [*time*] **-until** *date*	*date* is the date in the format *mm/dd/yy*. *time* (optional) is the time in the format [*hh:mm*[:*ss*]].	Limits the backup history display to a range of dates. Use **-since** to show the backups that occurred on or after a particular date or use **-until** to display the backup history that occurred on or up to a particular date.
-trailset primary | alternate		Displays the work items that were backed up by a primary or alternate set of media (trailset).
-level *levelnumber*	*levelnumber* is a level from 0 to 9, or a range of levels (for example, 0-8), for which you want to display backup history.	Displays all backups of the specified levels unless you use the other options to limit the report coverage.
-recent		Displays all of the work items that were backed up by EDM Backup, from the second most recent level 0 backup to the present. This report lists the standard **ebreport history** information.

Table 16-8 ebreport history Command Information (Continued)

Option	Argument Definition	Description
-volumes or -media		Displays the media volume names that are required to restore these backups.
-recover_size		Displays the amount of disk space that is required to restore each listed backup. The size is listed in KB, MB, GB, or TB as appropriate.
-seconds		Displays seconds in time reported.
-ebimport		Displays only backups that require ebimport(1m) before backup can be restored.
-expire_times		Displays all expire times (catalog, saveset, media), not just the one closest to expiration.
-completeness		Displays backup completeness mode for each listed backup saveset.
-dir		Displays the EBFS directory ID for each listed backup saveset.
-a or -baseline		Includes a baseline backup report after the history report.
-help		Displays command line options for **ebreport history**.
-nopartials		Skips backups that failed or are in progress.
-size		Displays the amount of data that was actually backed up. **Note:** The backup size that this report provides may be inconsistent with the recovery summary size. The algorithms that are used to calculate recovery size and backup size are different.

Sample Backup History Report

Figure 16-4 shows a sample report that is generated by:

```
#  ebreport history -recent
```

for an EDM called fig that backs up two templates (schedules): Generic and Server. The Server Template uses both a Primary media set (trailset) and an Alternate.

Figure 16-4 **EDM Backup History Report**

```
EDM Backup History Report for server edm at Sept 30 14:19:52 1998
Report options: -recent
**** Work Items for Template Generic, Primary Trailset ****
**Item "vigo:/work" for client "vigo"
Time           Lvl ID                 Status   Entries Expires Rcvr
10/19/98 16:12  0  72768A4B.32F65536 complete       210/20/98
10/13/98 16:04  0  72768A4B.32F65406 complete       210/14/98
:**** Work Items for Template Server, Primary Trailset ****
**Item "fig:/" for client "fig"
Time           Lvl ID                 Status   Entries Expires Rcvr
10/14/98 16:12  9  72768A4B.3329BF58 unsorted    405310/15/98 NO
10/ 6/98 10:40  0  72768A4B.331EE55C complete    525010/ 7/98
10/12/98 23:36  0  72768A4B.33029A9D complete    500710/17/98
.
.
**** Work Items for Template Server, Alternate Trailset ****

**Item "fig:/" for client "fig"
Time           Lvl ID                 Status   Entries Expires Rcvr
10/19/98 0:39  9  72768A4B.330A9265 complete 5045   10/23/98
10/13/98 16:35  0 72768A4B.33038955 complete 5006   10/17/98
.
.
10/30/98 20:00  0  72768A4B.32C9B7ED complete    432710/30/98
```

All backups with complete and delta listings are available for restoring and appear in the EDM Restore window.

Backup Disaster Reports

At the completion of every LOCAL_DATABASE backup (the backup of the EDM Backup database), the script /usr/epoch/EB/config/local_db_cleanup automatically generates a minimal disaster report. By default, this report is e-mailed to all EDM Backup administrators, appended to /usr/epoch/EB/config/disaster-report.log, and printed to the default system printer.

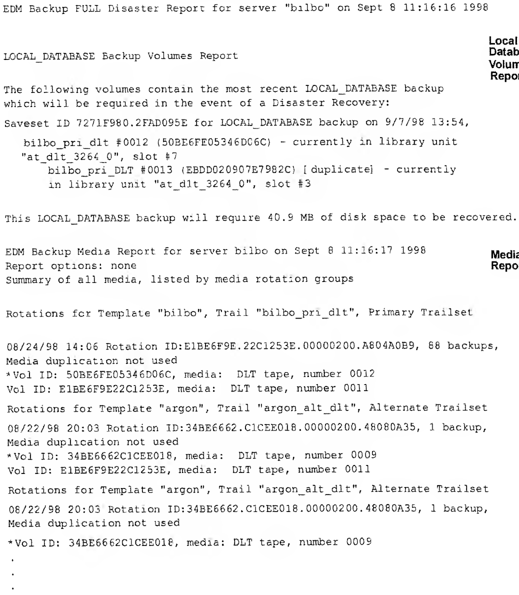
The minimal disaster report provides the essential information you need to perform a disaster recovery on the server. It is a subset of the full disaster report which is generated by the command **ebreport disaster**. You should run a full disaster report once every backup rotation and whenever significant system changes are made.

Refer to Chapter 19 ""Being Prepared for a System Disaster"" for instructions on preparing for a disaster. See Chapter 20 "Recovering a Server from a Disk Failure" and Chapter 21 "Recovering a UNIX Client from Disk Failure" for instructions on recovering a server and a client.

Figure 16-5 shows selections from the sections of the EDM Backup FULL disaster report. It contains the following sections:

- Local database volumes report
- Media report
- Backup history report
- Baseline backup history report (for HSM systems)
- Backup coverage report
- Backup installation report
- Backup configuration files
- List of installed clients
- Library Manager configuration data
- Filesystem table (/etc/vfstab)
- Locally mounted disks
- HSM local configuration
- The root crontab file

Figure 16-5 **EDM Backup FULL Disaster Report**



```
EDM Backup FULL Disaster Report for server "bilbo" on Sept 8 11:16:16 1998


LOCAL_DATABASE Backup Volumes Report


The following volumes contain the most recent LOCAL_DATABASE backup
which will be required in the event of a Disaster Recovery:

Saveset ID 7271F980.2FAD095E for LOCAL_DATABASE backup on 9/7/98 13:54,

   bilbo_pri_dlt #0012 (50BE6FE05346D06C) - currently in library unit
   "at_dlt_3264_0", slot #7
       bilbo_pri_DLT #0013 (EBDD020907E7982C) [duplicate] - currently
       in library unit "at_dlt_3264_0", slot #3


This LOCAL_DATABASE backup will require 40.9 MB of disk space to be recovered.


EDM Backup Media Report for server bilbo on Sept 8 11:16:17 1998
Report options: none
Summary of all media, listed by media rotation groups


Rotations for Template "bilbo", Trail "bilbo_pri_dlt", Primary Trailset


08/24/98 14:06 Rotation ID:E1BE6F9E.22C1253E.00000200.A804A0B9, 88 backups,
Media duplication not used
*Vol ID: 50BE6FE05346D06C, media:  DLT tape, number 0012
Vol ID: E1BE6F9E22C1253E, media:  DLT tape, number 0011

Rotations for Template "argon", Trail "argon_alt_dlt", Alternate Trailset

08/22/98 20:03 Rotation ID:34BE6662.C1CEE018.00000200.48080A35, 1 backup,
Media duplication not used
*Vol ID: 34BE6662C1CEE018, media:  DLT tape, number 0009
Vol ID: E1BE6F9E22C1253E, media:  DLT tape, number 0011

Rotations for Template "argon", Trail "argon_alt_dlt", Alternate Trailset

08/22/98 20:03 Rotation ID:34BE6662.C1CEE018.00000200.48080A35, 1 backup,
Media duplication not used

*Vol ID: 34BE6662C1CEE018, media: DLT tape, number 0009
 .
 .
 .
```

Local Database Volumes Report

Media Report

History Report

```
EDM Backup History Report for server bilbo at March 8 11:16:21 1998
Report options: -recent -completeness -dir

**** Work Items for Template argon, Primary Trailset ****

**Item "argon" for client "cheetah"

Time            Lvl IDStatus Entries Expires serverdb Completeness Directory ID

 9/ 7/98 17:58  0  7271F980.2FAD0B70 complete33075  9/12/98normal files
BDBE8F48.D50C5B38.0023CD00.F40D4A0A
 9/ 6/98 17:58  9  7271F980.2FABB9E6 delta 4  9/11/98   normal files
BDBE8F48.D50C5B38.0022D400.610AF0E0
 9/ 5/98 17:58  9  7271F980.2FAA685E delta 24  9/10/98   normal files
BDBE8F48.D50C5B38.0021C000.B20A1983
 9/ 4/98 17:58  9  7271F980.2FA916DA delta 1006  9/9/98normal files
BDBE8F48.D50C5B38.0020B300.FF0C66BE
 9/ 3/98 17:58  8  7271F980.2FA7C582 delta 24  9/8/98normal files
BDBE8F48.D50C5B38.001F4E00.FC0D0D5F


**** Work Items for Template argon, Alternate Trailset ****

**Item "argon" for client "cheetah"

Time          Lvl ID               Status   Entries Expires  serverdb Completeness
Directory ID

9/22/98 20:02  0  7271F980.2F999998 complete   32434 9/24/98   normal files
34BE6662.C1CEE018.00000500.4B0EB453
.
.
.
```

Baseline Backup History Report (for HSM systems)

```
EDM Backup Baseline Backup History Report for server fig at Sept 27 14:43:49 1998
Report options: none

**** Baseline Backups for Template Server, Primary Trailset ****

**Item "fig:/catalogs"

Time            Lvl ID                Status

 9/ 5/98 10:23 B1  72768A4B.32F8A612 partial


**Item "fig:/client_data1"

Time            Lvl ID                Status

 9/ 5/98 17:41 B1  72768A4B.32F90C9A no cat

 9/ 5/98 10:23 B1  72768A4B.32F8A611 no cat

 9/ 4/98 19:56 B1  72768A4B.32F7DADA partial
```

Coverage Report

```
EDM Backup Coverage Report for server adam at Sept 8 11:16:23 1998
Report options: none
Filesystems currently backed up:CurrentMaxCurrentMax
--------------------------------------------------------------------------------
adam:/ 4381/41536 files,50.3 MB/ 74.9 MB adam:/data1 27/ 63152 files,3.4 MB/  250.0
MB adam:/data2 21/     63152 files, 3.3 MB/250.0 MB
hamster:/ 17854/215040 files,600.1 MB/  778.2 MB negril:/ 3685/ 98176 files, 64.1
MB/  187.9 MB negril:/data 5/    191872 files, 1.3 MB/  750.8 MB
..
Total: 17 filesystems backed up 62480/ 2202176 files, 1.4 GB/    6.9 GB
```

Installation Report

```
EDM Backup Installation Report for server adam at Sept 27 14:23:26 1998
Report options: -all
.
.
.
EDM Backup currently running load 6.0.0.0

/usr/epoch IS A SYMLINK to /ep_usr/epoch
/usr/epoch/EB is a real directory under /ep_usr/epoch
/usr/epoch/GENDIR IS A SYMLINK to /home/epoch

/usr/epoch/EB/adam is a real directory under /usr/epoch/EB
/usr/epoch/EB/bin is a real directory under /usr/epoch/EB
/usr/epoch/EB/catalogs IS A SYMLINK to /home/epoch/EB/catalogs
.
.
The local client is of the type:   sun_sun4_v55_srv
The client backup username is:     ebadmin
The user  ID for ebadmin is:       24375
The group ID for ebadmin is:       25
The home directory for ebadmin is: /usr/epoch/EB
Client negril is of type sun_sun4_v55_emc (5.0.0.0) installed 1998 Sept 13 16:06:21
Client adam is of type sun_sun4_v55_srv (5.0.0.0) installed 1998 Sept 27 13:24:46
Client hamster is of type hp_700_v9 (5.0.0.0) installed 1998 Sept 27 13:55:29

End of EDM Backup Installation Report for server adam at Sept 27 14:23:26 1997
```

Backup Configuration Files

```
Displaying current EDM Backup configuration...

Sept 8 11:16 1998 /tmp/eb.cfg Page 1

ebserver: "bilbo"
{
client backup username: "ebadmin";

backup administrator usernames:
"root",
"gil";

authorized backup list:
"argon",
"bilbo",
.
.
.
startup parameters:
                {
                perform initial full backup as soon as possible;
                } /* end startup parameters */
      } /* end server block */.
.
.
.
```

Installed Clients

```
#Clients Installed File
#
#Format: I,len,host,timestamp,len,method,invocations,platform,RMS,vlength,version,;
#
I,  7,wildcat,837787783, 7,edmlink,0,99,0,  7,5.0.0.0,;
I,  4,fish,842992815,  7,netware,10,113,0,  7,5.0.0.0,;
I,  6,berlin,850748622,  7,netware,10,53,0,  7,5.0.0.0,;
I,  7,warthog,850776740,  7,netware,0,113,0,  7,5.0.0.0,;
I,  4,zero,850777296,  7,netware,0,114,0,  7,5.0.0.0,;
I,  7,hamster,851018343,  3,rsh,0,75,0,  7,5.0.0.0,;
I,  6,jumper,856192150, 7,edmlink,0,107,0,  7,5.0.0.0,;
I,  4,vigo,856385813, 7,edmlink,0,109,1,  7,5.0.0.0,;
I,  8,chipmunk,857155162,  3,rsh,1,97,0,  7,6.0.0.0,;
I,  6,negril,858032953, 7,edmlink,0,109,1,  7,6.0.0.0,;
I, 12,indianapolis,858725141,  7,netware,1,53,0,  7,6.0.0.0,;
I,  3,fig,859312730,  6,direct,0,108,1,  7,6.0.0.0,;
I,  6,bolton,859571785, 7,edmlink,0,93,0,  7,6.0.0.0,;
I,  7,pilgrim,859821588, 7,edmlink,0,99,0,  7,6.0.0.0,;
```

Library Manager Configuration

```
Displaying library manager configuration (used with lmconfig...)

* Lu_name         Name          ID                Status
L offline_0       -             -                 synced
L offsite_0       -             -                 synced
L at_dlt_3264_0   -             (0,1,1,0)         synced
D at_dlt_3264_0   drive_0       (0,1,5,0)         enabled
D at_dlt_3264_0   drive_1       (0,1,4,0)         enabled
D at_dlt_3264_0   drive_2       (0,1,3,0)         enabled
L hp_mf_c17xx_0   -             (0,2,6,0)         synced
D hp_mf_c17xx_0   drive_0       (0,2,5,0)         enabled
D hp_mf_c17xx_0   drive_1       (0,2,4,0)         enabled
D hp_mf_c17xx_0   drive_2       (0,2,2,0)         enabled
D hp_mf_c17xx_0   drive_3       (0,2,1,0)         enabled
```

Filesystem Table /etc/vfstab

```
Sept 21 10:51 1998 /etc/vfstab Page 1

#device          device          mount          FS      fsck    mount   mount
#to mount        to fsck         point          type    pass    at boot options
#
#/dev/dsk/c1d0s2 /dev/rdsk/c1d0s2 /usr ufs     1        yes     -
fd       - /dev/fd fd       -        no       -
/proc    - /proc   proc     -        no       -
/dev/dsk/c0t3d0s1 - -        swap     -        no       -
/dev/dsk/c0t3d0s0        /dev/rdsk/c0t3d0s0       /        ufs     1       no      -
/dev/dsk/c0t3d0s6        /dev/rdsk/c0t3d0s6       /usr     ufs     2       no      -
/dev/dsk/c0t2d0s3        /dev/rdsk/c0t2d0s3       /ep_usr  ufs     3       yes     -
/dev/dsk/c0t3d0s5        /dev/rdsk/c0t3d0s5       /home    vxfs    4       yes     -
/dev/dsk/c0t2d0s0        /dev/rdsk/c0t2d0s0       /data1   vxfs    5       yes     -
/dev/dsk/c0t2d0s1        /dev/rdsk/c0t2d0s1       /data2   vxfs    6       yes     -
/dev/dsk/c0t3d0s7        /dev/rdsk/c0t3d0s7       /home1   ufs     7       yes     -
swap     - /tmp    tmpfs    -        yes      -
```

Locally Mounted Disks

```
Displaying locally mounted disks...
/                  (/dev/dsk/c0t3d0s0 ):        8192 block size 1024 frag size
  153534 total blocks      50292 free blocks    34952 available 41536 total files
   37126 free files      8388632 filesys id
     ufs fstype        0x00000004 flag            255 filename length

/usr               (/dev/dsk/c0t3d0s6 ):        8192 block size 1024 frag size
  769694 total blocks     432834 free blocks   355874 available 192576 total files
  177771 free files      8388638 filesys id
     ufs fstype        0x00000004 flag            255 filename length
.
.
.
/home              (/dev/dsk/c0t3d0s5 ):        8192 block size 1024 frag size
  495936 total blocks     176534 free blocks   150656 available 23840 total files
   21507 free files      8388637 filesys id
    vxfs fstype        0x00000004 flag            255 filename length
```

HSM Local Configuration (HSM is optional with EDM)

```
EpochMigration Local Configuration
EpochMigration System Configuration:
    Enable_stage_out  Max_trails  Enable_self_describing
            Y              1             N

Staging trail "PubsTrail_1"
    Stage outs enabled: Y  Media: EO    Unrestricted
    Self-Describing enabled: N
    Enable  HWM  LWM  PSWM  Delay  Mntpoint
       Y     95   88   80       0  defaults for PubsTrail_1
       Y     95   88   80       0  /home
.
.
.
EpochMigration Current Migration Volumes
Current primary staging volumes are:

    Staging trail "PubsTrail_1"
    Sequence: 14  Mside: 1  Volid: 11CC39F59912E6A9  Nblocksavail: 314529

    Staging trail "PubsTrail_2"
    Sequence: 13  Mside: 1  Volid: 89CC2F69030965E2  Nblocksavail: 314529
```

root crontab

```
Displaying root crontab...
#ident"@(#)root1.1193/04/08 SMI"/* SVr4.0 1.1.3.1*/
#
# The root crontab should be used to perform accounting data collection.
#
0 2 * * 0,4 /etc/cron.d/logchecker
5 4 * * 6   /usr/lib/newsyslog
15 3 * * * /usr/lib/fs/nfs/nfsfind
# Invoke EDM Backup backup program #EPCebs
45 13 * * * /usr/epoch/EB/bin/ebbackup bilbo >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup backup program              #EPCebs
00 14 * * * /usr/epoch/EB/bin/ebbackup argon >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup backup program              #EPCebs
15 14 * * * /usr/epoch/EB/bin/ebbackup lucifer >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup catalog expiration program #EPCebs
30 00 * * * /usr/epoch/EB/bin/ebexpire -expire -purge >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup catalog cleanup program #EPCebs
00 1 * * * /usr/epoch/EB/bin/ebcatclean -fix_saveset >/dev/null 2>&1 #EPCebs
# Invoke EDM Backup LOCAL_DATABASE validity check program #EPCebs
00 3 * * * /usr/epoch/EB/config/local_db_warning  >/dev/null 2>&1 #EPCebs
#40 * * * * /usr/epoch/lib/epnewlog 500000 > /dev/null 2>&1#EPCgl
#00 23 * * 6 /usr/epoch/lib/epnewlog > /dev/null 2>&1#EPCgl
#00 07 * * * /usr/epoch/lib/eptrunclog root > /dev/null 2>&1#EPCgl
#30 08 * * * /usr/epoch/lib/epcleanup > /dev/null 2>&1#EPCgl
40 * * * * /usr/epoch/lib/epnewlog 500000 > /dev/null 2>&1#EPCgl
00 23 * * 6 /usr/epoch/lib/epnewlog > /dev/null 2>&1#EPCgl
00 07 * * * /usr/epoch/lib/eptrunclog root > /dev/null 2>&1#EPCgl
30 08 * * * /usr/epoch/lib/epcleanup > /dev/null 2>&1#EPCgl
#50 8 * * * /usr/epoch/lib/ebfs/ebfs_cleanup > /dev/null 2>&1#EPCebfs
50 8 * * * /usr/epoch/lib/ebfs/ebfs_cleanup > /dev/null 2>&1#EPCebfs
.
.
.
End of EDM Backup FULL Disaster Report for server "bilbo" on Sept 8 11:16:16 1998
```

End of EDM Backup Disaster Report

Backup Baseline Reports

In HSM systems, **ebreport baseline** generates the baseline report which presents a summary of backup baseline activity as reflected in the saveset database. A status line is printed for every non-expired baseline backup of every work item selected by the arguments.

Table 16-9 **ebreport baseline Command Information**

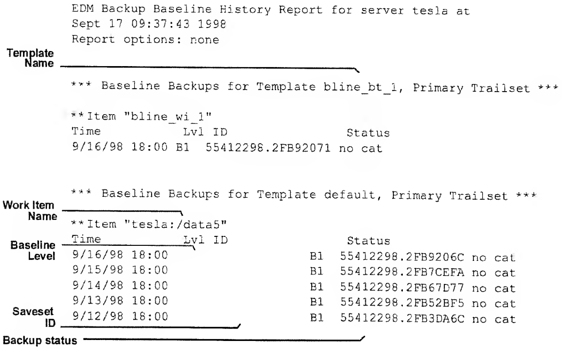
Option	Argument Definition	Description
-client *clientname*	*clientname* is the client whose baseline history you want to display.	Displays only backups that were created for the named client.
-item *workitemname*	*workitemname* is the client's work item for which you want to display baseline history.	Displays only backups that were created for the named work item. If the name "*" is used, all work items are selected. Note that "*" must be quoted on the command line.
-template *templatename*	*templatename* is the backup template for which you want to see a baseline summary.	Displays only backups that were run from the named template (schedule). If the name "*" is used, all work items are selected. Note that "*" must be quoted on the command line.
-completeness		Also displays the backup completeness mode for each reported work item backup.
-recent		Lists only baseline backups since the second most recent level 0 backup for each work item.
-since *date* [*time*] **-until *date***	*date* is the date in the format *mm/dd/yy*. *time* (optional) is the time in the format [*hh*:*mm*[:*ss*]].	Limits the backup display to a range of dates. Use **-since** to show the backups that occurred on or after a particular date or use **-until** to display the backup that occurred on or up to a particular date.

Table 16-9 **ebreport baseline Command Information (Continued)**

Option	Argument Definition	Description
-trailset primary | alternate		Displays the work items that were backed up by a primary or alternate trailset.
-level [B1 | B2]		Displays only savesets for backups of the given level. You can enter up to two **-level** options in a single invocation, each occurrence adding another level to the selection set.

Figure 16-6 shows a sample baseline report generated by **ebreport baseline**.

Figure 16-6 **EDM Backup Baseline Report**



Template Name
Work Item Name
Baseline Level
Saveset ID
Backup status

```
EDM Backup Baseline History Report for server tesla at
Sept 17 09:37:43 1998
Report options: none

*** Baseline Backups for Template bline_bt_1, Primary Trailset ***

**Item "bline_wi_1"
Time           Lvl ID                Status
9/16/98 18:00 B1  55412298.2FB92071 no cat


*** Baseline Backups for Template default, Primary Trailset ***

**Item "tesla:/data5"
Time           Lvl ID                Status
9/16/98 18:00                  B1  55412298.2FB9206C no cat
9/15/98 18:00                  B1  55412298.2FB7CEFA no cat
9/14/98 18:00                  B1  55412298.2FB67D77 no cat
9/13/98 18:00                  B1  55412298.2FB52BF5 no cat
9/12/98 18:00                  B1  55412298.2FB3DA6C no cat
```

Backup Completion Reports

EDM Backup prepares backup completion reports and can send them to specified individual(s) via a shell script. (For setup details see "Backup Completion Script" on page B-79.) Figure 16-7 shows a sample backup completion report.

Figure 16-7 **EDM Backup Completion Report**

Date, Time and Process ID #, Template

```
9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
Summary report for processing template "mwf"
```

Date, Time and process ID #, Work Item, Template, Trailset, Trail, Number of Files Backed up and Number of kb Used

```
9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
processing of work item "cad1-all" via template "default" Level 0
SUCCEEDED
trailset was "cdr", trail was "cdr tape", 59 files backed up in
886KB
9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
processing of work item "cad2-all" via template "default"
SUCCEEDED
trailset was "cdr", trail was "cdr tape", 312 files backed up
in 55809KB
9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
processing of work item "cad3-all" via template "default"
SUCCEEDED
trailset was "cdr", trail was "cdr tape", 10257 files backed up
in 135514KB
9/05/98 18:31:21 [ 2295:/usr/epoch/EB/bin/ebbackup]
processing of work item "cad4-all" via template "default"
SUCCEEDED
trailset was "cdr", trail was "cdr tape", 25306 files backed up
in 203749KB
```

The server **eb_server_config** installation procedure creates the **mailok** script to which it passes the backup completion information. The script mails a completion statement to individuals responsible for backup operations, and/or writes them to a log.

Because the **ebbackup** program mails these reports immediately after a backup, you can read them as soon as the backup completes.

Backup Failure Reports

Whenever EDM Backup encounters an error that prevents backup completion (for example, a client system has crashed), it generates a backup failure report. (For setup details see "Backup Failure Script" on page B-80.)

The EDM Backup program can send backup failure reports to specified individuals via a shell script. Because EDM Backup mails these reports whenever a failure occurs, you are notified of a failure as soon as it happens. On the other hand, if you don't receive one of these reports, you can assume your backups are successful. When you receive a backup failure report, you should fix the problem with the client system. However, EDM Backup continues to back up all other clients in the backup template (schedule), skipping those that had a problem.

Figure 16-8 shows a sample backup failure report.

Figure 16-8 **EDM Backup Failure Report**

Date, Time, Process ID #, Error Number, Work Item Name, and Reason for Failure

```
9/06/98 06:22:21 [ 3423:ebbackupd errno=35[Operation would
block] , ec=0x19] Workitem "doc1-all" backup TIMED-OUT
```

The server **eb_server_config** installation procedure creates the **mailerr** script to which it passes the backup failure information. The script mails a failure statement to individuals who are responsible for backup operations, and/or writes them to a log.

Backup Coverage Reports

The **ebreport coverage** command makes it easy for you to determine if new filesystems were added to client systems and if they are getting backed up. When the report lists a filesystem that EDM Backup is not currently backing up, and the filesystem is one that you want to backup, you'll need to edit the client's work item statement to add the filesystem to the list of backup files.

> **Note:** **ebreport coverage** reports on Unix and Windows NT filesystems only (no NetWare filesystems).

You can use the **ebreport coverage** command to display backed up and non-backed up filesystems on EDM Backup clients. The **ebreport coverage** command displays the backup status of all filesystems or you can use the options to display the following information.

Table 16-10 **ebreport coverage Command Information**

Command	Argument Definition	Description
-client *clientname*	*clientname* is the client whose backup history you want to display.	Displays a single client's backed up and non-backed up filesystems.
-completeness		Shows what kind of data is being backed up (for resident files only).
templatename	*templatename* is the backup template (schedule) for which you want to display backup history.	Displays a backup template's non-backed up filesystems; with the *templatename* option displays the backup status of the specified template(s') filesystems.
-installed		Shows installed EDM Backup clients – displays all backed up and non-backed up filesystems in installed client list.

Figure 16-9 shows a report generated by **ebreport coverage** for three clients (adam, hamster, and negril), and identifies the fields in the report.

Figure 16-9 **EDM Backup Coverage Report**

```
EDM Backup Coverage Report for server adam at Sept 8 11:16:23 1998
Report options: none


Filesystems currently backed up:Current      Max          Current      Max
-------------------------------------------------------------------------------
adam:/                          4381/        41536 files, 50.3 MB/     74.9 MB
adam:/data1                     27/          63152 files, 3.4 MB/      250.0 MB
adam:/data2                     21/          63152 files, 3.3 MB/      250.0 MB
hamster:/                       17854/       215040 files, 600.1 MB/   778.2 MB
negril:/                        3685/        98176 files, 64.1 MB/     187.9 MB
negril:/data                    5/           191872 files, 1.3 MB/     750.8 MB
negril:/data1                   4/           191872 files, 1.3 MB/     750.8 MB
.
.
.
Total: 17 filesystems backed up 62480/       2202176 files, 1.4 GB/     6.9 GB
```

Volume Reports

The **dbreport *reportname*** command generates reports from the system administration database.

Note: Ordinarily, only privileged users (those running as root) can run **dbreport**.

Some of these reports are listed in Table 16-11, to see a full list of reports see the **dbreport** man page.

Table 16-11 **dbreport Command Information**

Report Name	Description
volume	Generates a report of all the volumes known to the system. The fields of the report describe the type of media, the name of the application that currently owns the volume, the name assigned to the volume, the sequence number of the volume. the side of the volume, and the barcode of the volume, if any.
available	Generates a report of just those volumes that are currently available for allocation.
appl_usage	Generates a report of application volume usage statistics. The fields of the report are the application name. the volume name, the media type. the sequence number, the side, barcode, the number of blocks available, used. and stale in 1 KB units. the percentage of the volume which is stale data, and the number of files used and stale on the volume.
online	Generates a report of all media in system library units. The report is sorted by application, media type and application-dependent name.
offline	Generates a report of all media not in any system library unit. The report is sorted by application, media type and application-dependent name.
offsite	Generates a report of all media which has been moved to offsite storage. The volumes in this category can be re-introduced to the system by being injected into a library unit. The report is sorted by application, media type and application-dependent name.

Table 16-11 **dbreport Command Information (Continued)**

Report Name	Description
hsm	Generates a report of staging volume usage statistics. The fields of the report are the volume name, sequence number, side, barcode, the number of blocks used and stale in 1KB units, the percentage of the volume which is stale data, and the number of files used and stale on the volume.
baseline	Generates a report of baseline volume usage statistics. The fields of the report are the volume name, sequence number, side, barcode, the number of blocks used and stale in 1KB units, the percentage of the volume which is stale data, and the number of files used and stale on the volume
compaction	Generates a report you can use to estimate the staleness of volumes in order to determine which HSM volumes to compact with the **emcompact** utility.

Log Files

You can access backup log files directly and monitor them or review them for troubleshooting. For example, use **tail -f** to monitor progress during processing and use **vi** or other editor to review logs at a later time.

When filled, the oldest ten percent of these files is deleted on an ongoing basis.

Server Log Files

EDM Backup automatically creates log files in the directory /usr/epoch/EB/log on the EDM Backup server.

- The backups.log file contains information about backup operations. EDM Backup adds information to this file each time it backs up a template's work items. Selected notifications that appear in this log file also appear in other backup reports. It accumulates detailed shutdown and startup information each time a database work item is backed up.

Every two minutes **ebbackup** reports the average rate in KB/s for ALL work items being backed up. This rate is affected by process timing, data buffering, and overhead in **ebbackup**. If you want to see the rate for a specific tape drive, see the EDM Library Unit Manager window which reports on an active drive every thirty seconds.

- The recoveries.log file contains file restore startup and completion notifications. Use these files for comprehensive information about backup and restore activities on your EDM server.
- The ebcat.log contains startup and shutdown times for catalog processing and output from **ebexpire** and **ebimport**.
- The *template_name*.log records backup information for a single template.

  Whenever you want to see the backup history of a single backup template (schedule), use the *template_name*.log file. The information in this log file varies depending on the *logging level* you specified in the configuration database (see "Server Log File" on page B-78). Thus, you can use the file to view a history of backup-related events for a single template.

  The default log file is located in /usr/epoch/EB/log/default_template.log.

  The default logging level, *stats*, reports when each client backup or restore begins, and includes periodic progress indications.

  There are five logging levels. Use the *debug* and *per file* levels to diagnose problems only when instructed by customer service.

Local Client Log Files

EDM Backup automatically creates two log files on the EDM Backup client: the backups.log and recoveries.log in the directory /usr/epoch/EB/CLIENT_HOME/*client*. The backups.log file contains an audit trail of backup-related activities listed in chronological order. The recoveries.log file contains an audit trail of restore-related activities listed in chronological order.

- backups.log accumulates detailed scanning information each time a local work item is backed up.
- recoveries.log records general start and end notifications for restore processing each time a local work item is restored.

Remote Client Log Files

On the remote clients, backups.log and recoveries.log files reside in the directory /usr/epoch/EB/CLIENT_HOME/*clientname*. They record network backups and restore operations.

Other Logs

Other logs record volume management and other system activity:

- System logs are located in /var/adm
- System logs are archived in /usr/epoch/adm
- Circular logs are located in /usr/epoch/adm/circular

System logging is configurable in /etc/syslog.conf.

Part IV
Command Line Interfaces

17 Configuring Library Managers

When you change EDM configuration by adding or removing a library unit, you need to reconfigure the software to recognize the change.

This chapter describes the script that you use to install device drivers and configure Library Managers for library units that are connected to the EDM.

The chapter describes the following tasks:

- Using the lmconfig Utility
- Listing Library Managers
- Installing Device Drivers
- Updating Device Drivers
- Removing Device Drivers
- Configuring a Library Manager
- Deconfiguring a Library Manager

Using the lmconfig Utility

The **lmconfig** utility, enables you to:

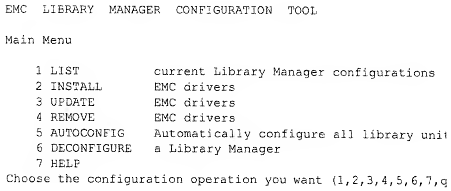
- list currently configured Library Managers
- install, update, and remove device drivers
- configure and deconfigure Library Managers
- access a help option that briefly describes the main menu entries

(Refer to the lmconfig man page for more information about this utility.)

Note: lmconfig is located in /usr/epoch/bin. Make sure that this pathname is defined in your PATH environment variable.

To start the configuration script, log in as root and enter the following command to display the main menu. In this menu, you select the configuration you want to perform.

```
# lmconfig
```



```
EMC  LIBRARY  MANAGER  CONFIGURATION  TOOL

Main Menu

    1 LIST           current Library Manager configurations
    2 INSTALL        EMC drivers
    3 UPDATE         EMC drivers
    4 REMOVE         EMC drivers
    5 AUTOCONFIG     Automatically configure all library unit
    6 DECONFIGURE    a Library Manager
    7 HELP
Choose the configuration operation you want (1,2,3,4,5,6,7,q
```

Listing Library Managers

When you choose 1 LIST from the main menu, Library Managers that are currently configured in /usr/epoch/etc/lm appear. lmconfig lists the name of the Library Manager and the device's SCSI address for the robot and each drive, as shown in the following example:

```
lmconfig completed configurations:

offline_0
offsite_0
at_452_0
   r0: 0 2 2 0
   d0: 0 2 3 0
   d1: 0 2 4 0
   d2: 0 2 5 0
```

Library Manager Name

The Library Manager's name identifies the manufacturer, drive type, and model number of the device.

Note: In releases previous to EDM 4.5.0, the Library Manager name includes the drive type (DLT, DTF, HITC, etc.); for example, "at_dlt_452_0."

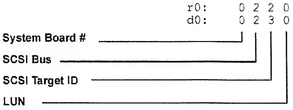
For example, the Library Manager name, at_452_0, has the following meaning:

Table 17-1

at	Manufacturer of the automated tape library unit: ATL Products.
452	Manufacturer's model number; in this example, the ACL 4/52 automated tape library unit. (4 drives/52 slots)
_0	Indicates the first Library Manager that is configured for this library unit type. The suffix increments for each additional library unit of this type that you configure.

SCSI Address

The SCSI address includes the system board number, SCSI bus, SCSI target ID, and logical unit number (LUN) of the library unit robot and drive(s); for example:



```
r0:     0 2 2 0
d0:     0 2 3 0
```

In this example, the library unit's robot and drive are on system board 0, SCSI bus 0; the robot's SCSI target ID is 0, and its LUN is 0. The library unit has one internal drive at SCSI target ID 1, LUN 0.

Installing Device Drivers

When you choose 2 INSTALL from the main menu, lmconfig installs the device drivers into the /devices directory. You must install device drivers before you configure a Library Manager. This option requires that at least one library unit be connected to the server and operational.

To install device drivers, use the following procedure.

1. Choose 2 INSTALL from the main menu. A prompt asks you to confirm the installation:

```
Main Menu
    1 LIST          current Library Manager configurations
    2 INSTALL        EMC drivers
    3 UPDATE         EMC drivers
    4 REMOVE         EMC drivers
   5 AUTOCONFIG   Automatically configure all library units
    6 DECONFIGURE    a Library Manager
    7 HELP
Choose the configuration operation you want (1,2,3,4,5,6,7,q)? 2
About to install all EMC drivers.
Do you wish to continue (y,n)? y
```

2. Enter **y** to begin driver installation. (Note that driver names vary by platform.) The script displays several messages that confirm driver installation.

```
Modifying kernel driver.conf files
Modifying /kernel/drv/st.conf

Installing drivers
Installing driver mo
Driver mo installed
Installing driver sjb
Driver sjb installed
```

Note: Ignore messages that indicate failure of mo driver installation.

3. After the drivers are installed, shut down the system to the PROM level by entering the following command:

   ```
   # shutdown -y -i6 -g0
   ```

   This command enables a reconfiguration reboot of the server.

4. After EDM shuts down and reboots, log in as root and restart lmconfig.

The main menu appears, as shown:

```
EMC  LIBRARY  MANAGER  CONFIGURATION  TOOL

Main Menu

    1 LIST           current Library Manager configurations
    2 INSTALL        EMC drivers
    3 UPDATE         EMC drivers
    4 REMOVE         EMC drivers
    5 AUTOCONFIG     Automatically configure all library uni
    6 DECONFIGURE    a Library Manager
    7 HELP
Choose the configuration operation you want (1,2,3,4,5,6,7,
```

EDM probes the bus for all attached hardware and assigns device nodes in the filesystem that represent the devices that are found. It also configures the logical namespace in /dev and the physical namespace in /devices.

5. Select 5 AUTOCONFIG to configure Library Managers automatically for the attached library units. Refer to "Configuring a Library Manager" on page 17-8 for this procedure.

Updating Device Drivers

Choose 3 UPDATE from the lmconfig menu to reinstall device drivers. You must update the device drivers after updating the EDM software or updating the module that contains the drivers.

When you update device drives, no Library Manager reconfiguration is required.

To update the device drivers:

1. Choose 3 UPDATE from the main menu.
2. Confirm the update at the prompt.

Removing Device Drivers

Choose 4 REMOVE from the **lmconfig** menu to remove all device drivers from the /devices directory. You must remove device drivers before deinstalling the EDM software.

To remove device drivers, select 4 REMOVE from the main menu. Then confirm the removal at the prompt.

Sample output appears below.

```
Main Menu

    1 LIST          current Library Manager configurations
    2 INSTALL       EMC drivers
    3 UPDATE        EMC drivers
    4 REMOVE        EMC drivers
   5 AUTOCONFIG   Automatically configure all library units
    6 DECONFIGURE    a Library Manager
    7 HELP
Choose the configuration operation you want
(1,2,3,4,5,6,7,q)? 4
About to remove all EMC drivers.
Do you wish to continue (y,n)? y

Modifying kernel driver.conf files
Modifying /kernel/drv/st.conf

Removing drivers
Removing driver mo
Removing driver sjb
```

Configuring a Library Manager

Use the AUTOCONFIG option of lmconfig to configure a Library Manager automatically for each library unit that is attached to the EDM.

AUTOCONFIG verifies that offline and offsite daemons are configured. Then it searches for all device nodes in the system, acquires all necessary information, and configures a library manager for each library unit. AUTOCONFIG automatically unloads all drives and puts the media into empty slots.

Because the operation is automatic, you do not have to know the system board numbers, SCSI bus numbers, target IDs, and LUN numbers for each device.

The lmconfig utility creates a subdirectory in /usr/epoch/etc/lm, copies a sample configuration file into the directory and modifies it, creates a link to the executable file, and adds the pathname of the new Library Manager to the Volume Manager's configuration file. (See Appendix C "Volume Management Configuration Files" for more information about how the Volume Manager uses the vm.cfg file.)

Note: You can also use enhanced lmconfig to run AUTOCONFIG and tell it not to ask any questions. If it detects any drives with media in them, AUTOCONFIG unloads the media automatically without asking you, and configures all unconfigured library units. You run enhanced lmconfig by using the command **lmconfig -A**.

Preparing for Configuration

Before you configure Library Manager(s), verify that:

- your hardware configuration is valid
- library units are properly cabled, powered up, and online
- device drivers were installed successfully and EDM rebooted
- all library unit drives are operational
- at least one piece of media is loaded in each library unit

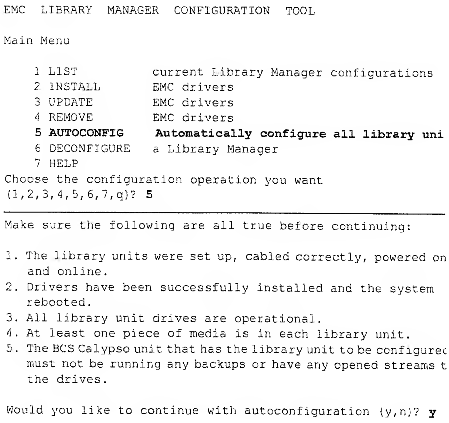
Running lmconfig

To start configuration, do the following:

1. Be sure you are logged in as root.
2. Enter the following command to display the lmconfig main menu:

   ```
   #  lmconfig
   ```

3. Choose 5 AUTOCONFIG from the main menu and then confirm autoconfiguration at the prompt.



```
EMC  LIBRARY  MANAGER  CONFIGURATION  TOOL

Main Menu

    1 LIST          current Library Manager configurations
    2 INSTALL       EMC drivers
    3 UPDATE        EMC drivers
    4 REMOVE        EMC drivers
    5 AUTOCONFIG    Automatically configure all library uni
    6 DECONFIGURE   a Library Manager
    7 HELP
Choose the configuration operation you want
(1,2,3,4,5,6,7,q)? 5
```

```
Make sure the following are all true before continuing:

1. The library units were set up, cabled correctly, powered on
   and online.
2. Drivers have been successfully installed and the system
   rebooted.
3. All library unit drives are operational.
4. At least one piece of media is in each library unit.
5. The BCS Calypso unit that has the library unit to be configurec
   must not be running any backups or have any opened streams t
   the drives.

Would you like to continue with autoconfiguration (y,n)? y
```

If All Library Units Are Configured

AUTOCONFIG looks for unconfigured devices, active library units, and loaded drives. If all library units are already configured, the following appears:

```
==========================================
        Starting Autoconfig v3.0
==========================================

Determining unconfigured devices:    100%
Searching for active library units:  100%
_autoconfig failed: *** Error: No non-configured library units found

autoconfig: No configuration done

lmconfig warning: AUTOCONFIG failed
```

Nothing was done because all available library units are already configured.

If Media Is Found In Any Drive

If AUTOCONFIG detects media in a drive it automatically unloads the drive and places the media into an empty slot. (Sample output follows.)

Select **y** (yes) to continue with configuration.

Note: If a mechanical problem does not allow the media to be moved, AUTOCONFIG fails. If you cannot fix the problem, shut off the problem library unit and then run AUTOCONFIG again to configure the remaining library units.

```
==========================================
          Starting Autoconfig v3.0
==========================================

Determining unconfigured devices:    100%
Searching for active library units:  100%
Checking for loaded drives:          100%

*********************************************************
* Found some drives loaded with media.
* Unconfigured drives must not have media in them.
*
* WARNING:  The unload program will unload all
*           unconfigured drives attached to this server.
*********************************************************

Would you like to unload the drive(s)  (y,n)? y

Searching for loaded drives in library unit:
  Vendor[ HP] Product[ C1710T] :: Board Bus Target LUN [ 0 4 0 0]

  Found 1 drives loaded with media
  Unloading.........

  Moving media from drive 0 to slot 6
```

The Configuration Process

AUTOCONFIG displays a list of available library units that are not yet configured. At the prompt, enter "**a**" for all, or a comma-separated list to select some but not all of the listed library units.

AUTOCONFIG now configures all or selected library units.

```
Please choose which library units to configure :
_______________________________________________

1.Vendor[ HP] Product[ C1710T] :: Board Bus Target LUN [ 0 4 0 0]

Please enter a comma separated list of the library
units you would like to configure, or 'a' for all :  a

::::::::::::::::::::::::::::::::::::::::::::::::::::::

Getting Info on Library Unit #0 : HP : C1710T : 6.10
The robot on library unit 1 is located at:
=== BOARD:0   BUS:4    TARGET:0   LUN:0 ===
Number of drives = 2
Number of slots  = 32
Number of inlets = 1
Media found in slot number 0
Library unit supports drive to drive moves
Using compatible configuration files:  hp_c17xx.attr / hp_c17xx.
=  =  =  =  =  =  =  =  =  =  =  =  =  =  =
Loading Drive 0. Please Wait......
Waiting for drive to be ready ..
Found Drive 0 : HP : C1716T : 3336
For your information, Drive 0 is located at :
=== BOARD:0   BUS:4    TARGET:4   LUN:0 ===
Using compatible configuration file for the drive:  eo_worm.tmpl
=  =  =  =  =  =  =  =  =  =  =  =  =  =  =
Loading Drive 1. Please Wait......
Waiting for drive to be ready ...
Found Drive 1 : HP : C1716T : 3336
For your information, Drive 1 is located at :
=== BOARD:0   BUS:4    TARGET:5   LUN:0 ===
Using compatible configuration file for the drive:  eo_worm.tmpl
::::::::::::::::::::::::::::::::::::::::::::::::::::::
Configuring library unit .........
Enter physical location for LU qntm_x700_0 : B1 Lab
```

Enter the physical location of the library unit; for example, "B1 Lab" as shown above.

Selecting the Cleaner Barcode Default

AUTOCONFIG prompts you to accept or decline a default barcode for cleaning cartridges:

```
Would you like to accept the EDM default
barcode of CLNXXX for tape cleaners (y,n)[ Y]? y
```

If you answer **y** (yes), the Library Manager recognizes any tape with a barcode of CLN(000-999) as a cleaner by default and does not place it in the drive during an inventory.

> **Note:** To override this default you must answer **n** (no).

Viewing Log Files

After all library units are configured, lmconfig notifies you that configuration is complete. AUTOCONFIG then asks if you want to view the log files. If you answer yes, information for each library unit appears:

```
Finished configuring library unit(s).

Would you like to see the log files (y,n)? y

============================
=     Library Unit Info
= Vendor ID   =  HP
= Product ID  =  C1710T
= Product Rev =  6.10
============================
( Device name : Board/Bus/Target/LUN : Vendor ID :
Product ID : REV )
Robot 0 : 0 4 0 0 : HP : C1710T : 6.10
Drive 0 : 0 4 4 0 : HP : C1716T : 3336
Drive 1 : 0 4 5 0 : HP : C1716T : 3336

::::::::::::::::::::::::::::::::::::::::::::::::::
```

Completing lmconfig

After the library managers are configured, type **q** to exit the lmconfig utility.

```
Main Menu

  1 LIST       current Library Manager configurations
  2 INSTALL      EMC drivers
  3 UPDATE       EMC drivers
  4 REMOVE       EMC drivers
 5 AUTOCONFIG Automatically configure all library units
  6 DECONFIGURE a Library Manager
  7 HELP

Choose the configuration operation you want
(1,2,3,4,5,6,7,q)? q
```

Reboot the EDM system by entering the following:

```
# shutdown -y -i6 -g0
```

This important step starts the vmdaemon, and ebfsd and vmdupd daemons. EDM software does not run until these daemons start.

A full inventory begins after the reboot; the time period for completing an inventory depends on the amount of media that the library unit contains.

Deconfiguring a Library Manager

When you deconfigure a Library Manager, lmconfig deletes the Library Manager's subdirectory and its contents. You should deconfigure a Library Manager when you permanently remove a library unit from the server.

To deconfigure a Library Manager, do the following:

1. Start lmconfig and choose 6 DECONFIGURE in the main menu.

2. lmconfig lists the currently installed Library Managers; at the prompt, select the one you want to remove:

```
Choose one or more Library Manager configurations to be removed
  1 offline_0
  2 offsite_0
  3 at_452_0
  4 hp_cl7xx_0
Enter comma-separated choice(s) on a single line (1, 2, 3, 4, q)
```

3. Enter the number(s) that correspond to the Library Manager(s) that you want to deconfigure (as shown in the example above).

   Note: If you inadvertently remove the offline or offsite Library Manager, lmconfig automatically adds it back for you. Just choose the CONFIGURE option in the lmconfig main menu.

```
offsite_0 removed

qntm_x700_0 removed

hp_cl7xx_0 removed

offline_0 removed
```

The utility removes the Library Manager's subdirectory and its contents from /usr/epoch/etc/lm. It also deletes the Library Manager from the vm.cfg file and notifies the Volume Manager to reread the file and kill the associated LM daemon.

4. When the main menu appears, select **q** (quit) to return to the system prompt.

```
Main Menu

    1 LIST          current Library Manager configurations
    2 INSTALL       EMC drivers
    3 UPDATE        EMC drivers
    4 REMOVE        EMC drivers
    5 CONFIGURE     Manually configure a library unit
    6 AUTOCONFIG    Automatically configure all library units
    7 DECONFIGURE   a Library Manager
    8 HELP
Choose the configuration operation you want (1,2,3,4,5,6,7,8,q)?
```

If You Have Trouble Configuring a Library Unit

If a drive other than drive 0 fails while configuring a library unit, AUTOCONFIG asks whether you want to configure the LU with the drives that AUTOCONFIG was able to find (this number is less than the total number of drives that the LU contains). Messages that are similar to the following appear:

```
*** ERROR:  Cannot find node for drive 1
***         Drive 1 may have been loaded using a
***         cleaning cartridge, or it may be damaged or disabl
***         Configuration of library unit #0 has stopped.

AUTOCONFIG could only find the first 1 drive(s)

Would you like to configure the library unit
with only 1 drive(s) (y,n)[N]? no
```

If you enter **n** (no), configuration of the library unit stops:

```
_autoconfig failed: *** Error: SCSI location of
drive #1 was not found

*** Error: Unable to configure library unit #0
```

If you enter **y** (yes), configuration completes with the drives that AUTOCONFIG found.

```
NOTE: There is a tape in drive 1

Please shut down EDM server after AUTOCONFIG finishes,
and manually remove the tape from the drive.
```

If a Problem Occurs While Configuring Multiple Library Units

If you want to configure more than one library unit at one time and AUTOCONFIG fails because a volume is stuck in a drive, AUTOCONFIG may have a problem while removing the volume from the drive.

Turn off the library unit, remove the volume from the drive, and check the drive cables. Then restart EDM and restart AUTOCONFIG. This enables AUTOCONFIG to configure the remaining library units.